

# ArchiMate<sup>®</sup> in de praktijk

*Modelleren volgens ArchiMate aan de hand van  
een verzameling good practices*

---



## Colofon

Titel : ArchiMate in de Praktijk

Datum : 2 februari 2012

Versie : 4.0

Verandering : T.o.v. 3.0: Nieuwe good practices toegevoegd, tekstuele verbeteringen gedaan en figuren qua stijl gelijk getrokken.

Status : Definitief

Redacteur : Hans van Drunen, Egon Willemsz

Bedrijf : Werkgroep ArchiMate Gebruik/Tools

Auteurs : Harmen van den Berg, Alexander Bielowski, Gertjan Dijk, Hans van Drunen, Gert Faber, Jan van Gijsen, Rienk de Kok, Ger Oosting, Wim Schut, Frans Smit, Jan van de Wetering, Egon Willemsz  
Eerdere versies: Hans Bosma, Frank Langeveld, Joost Luijpers, Robert Slagter



# Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>1</b>
1.1	Aanleiding	1
1.2	Doel en doelgroep	2
1.3	Leeswijzer	2
1.4	Wijzigingen ten opzichte van versie 3.0	3
1.5	Website	3
1.6	Oproep voor nieuwe good practices	3
<b>2</b>	<b>Modelleren in de bedrijfslaag</b>	<b>5</b>
2.1	Overzicht good practices	5
2.2	Bedrijfsservice, bedrijfsfunctie en bedrijfsproces	5
2.3	Afbakening bedrijfsservice	7
2.4	Afbakening bedrijfsproces	8
2.5	Afbakening bedrijfsfunctie	8
2.6	Bedrijfsproces en bedrijfsinteractie	9
2.7	Viewpoints voor procesmodellering	11
2.8	Samenwerking tussen organisaties	12
2.9	Informatiedomein	13
<b>3</b>	<b>Modelleren in de applicatielaag</b>	<b>15</b>
3.1	Overzicht good practices	15
3.2	Applicatielandschap	15
3.3	Koppeling tussen applicaties	17
3.4	Functionaliteit van een applicatie	18
3.5	Websites als applicatiecomponent en het gebruik van services	20
<b>4</b>	<b>Modelleren in de infrastructuur- / technologielaag</b>	<b>22</b>
4.1	Overzicht good practices	22
4.2	Infrastructuurservices	22
4.3	Infrastructuurlandschap	23
4.4	Onderscheid systeemsoftware en artefact	24
<b>5</b>	<b>Modelleren over de grenzen van lagen heen</b>	<b>25</b>
5.1	Overzicht good practices	25
5.2	Veelgebruikte relaties	25
5.3	Product en applicatieservices	30
5.4	Ondersteuning van batch en online processen	30
5.5	Servicebroker	31
5.6	Domein	33
5.7	Interne en externe services	34
5.8	Noodzaak voor het gebruik van services	36
5.9	Vereenvoudigingen met behoud van consistentie	37
5.10	Groepering	38
5.11	Gegevensuitwisseling met externe organisaties in applicatielaag	40
5.12	Weergeven van berichten in wachtrijen	42
5.13	'Nesten' en impliciet afgeleide relaties	44
5.14	File transfer	46
	<b>Referenties</b>	<b>48</b>



# 1 Introductie

## 1.1 Aanleiding

Hoe breng ik een applicatielandschap in beeld? Hoe leg ik relaties tussen logische (inrichtingsonafhankelijk) en fysieke bedrijfsprocessen (inrichtingsafhankelijk)? Hoe breng ik in beeld welke afdelingen functioneel eigenaar zijn van welke applicaties? Hoe beeld ik de applicatieportfolio af op de technische infrastructuur? Hoe geef ik aan welke gegevens benodigd zijn bij de levering van welke producten en diensten aan klanten? Hoe modelleer ik een servicebroker?

Dit zijn allemaal modelleervragen waar je als enterprise architect in de praktijk mee geconfronteerd wordt. Er zijn dan ook veel manieren die worden gehanteerd om antwoord te geven op deze vragen. Veel manieren leidt tot wildgroei en problemen om met de gemaakte platen goed te kunnen communiceren. Achter elk plaatje zit een verhaaltje, maar een plaatje gaat vaak een eigen leven leiden en het verhaaltje raakt verloren, zeker als de oorspronkelijke maker met andere werkzaamheden bezig gaat of zelfs weg is bij een organisatie.

In de praktijk van ‘werken onder architectuur’ wordt bij steeds meer organisaties gebruik gemaakt van ArchiMate als beschrijvingstaal voor het vastleggen van enterprise architecturen met daarin domeinen als: producten/diensten, processen, organisatie, gegevens, applicaties en technische infrastructuur. Met deze taal is het mogelijk om de globale structuur *binnen* een domein te modelleren en de relaties *tussen* domeinen te modelleren. Een andere belangrijke eigenschap van ArchiMate is de mogelijkheid om modellen op verschillende manieren te visualiseren, gericht op verschillende belanghebbenden. Bovendien levert de taal een formele onderbouwing zodat modellen geanalyseerd kunnen worden (bijvoorbeeld op een aspect als performance). Daarbij is het expliciet niet de bedoeling dat ArchiMate de bestaande talen, zoals BPMN, UML e.d., zal vervangen; het vormt hierop een aanvulling, waarbij wel zo veel mogelijk aangesloten wordt op bestaande standaarden en praktijken.

Ondertussen zijn er veel ervaringen opgedaan met ArchiMate in Nederland en andere landen en is te zien dat steeds meer herkenbaardere en beter communiceerbare modellen worden opgeleverd. En de populatie van enterprise architecten dat gebruik maakt van ArchiMate is nog steeds groeiende.

Hierbij is echter ook te zien dat dezelfde typen modelleervraagstukken op verschillende manieren met ArchiMate worden ingevuld, waarbij kan worden gesteld dat niet alle manieren een goede bijdrage leveren aan de communiceerbaarheid van de modellen richting de belanghebbenden. De praktijk leert ook dat je van goede huize moet komen om de taal goed te kunnen vastpakken en toepassen. De drempel is toch nog vrij hoog om er direct mee aan de slag te kunnen gaan.

## 1.2 Doel en doelgroep

De praktijk heeft behoorlijk wat vragen opgeleverd richting het ArchiMate Forum over de wijze van toepassing van de ArchiMate concepten.

Doel van voorliggend document is om een begin te maken met de beantwoording van deze vragen door middel van het aanreiken van beproefde oplossingen in de vorm van good practices, waarmee de drempel voor de toepassing van ArchiMate in de praktijk aanzienlijk kan worden verlaagd en de effectiviteit van de modellen die worden gemaakt kan worden vergroot.

Dit document is geschreven voor enterprise architecten die ArchiMate in de praktijk willen toepassen als taal voor het vastleggen van modellen van organisaties, de ICT-ondersteuning en de onderlinge relaties daartussen.

Dit document moet worden gezien als aanvulling op de bestaande ArchiMate-resultaten, waaronder (zie de bijlage voor een verwijzing naar deze referenties):

- Enterprise Architecture At Work.
- Inleiding in de ArchiMate-taal.
- Architecture Language Reference Manual.
- Viewpoints Functionality and Examples.
- ArchiMate Quick Reference.
- ArchiMate 1.0 Specification.
- ArchiMate 2.0 Specification.

Verder wordt aanbevolen:

- Kanaalpatronen (Kanalen in Balans).

## 1.3 Leeswijzer

De hierna volgende hoofdstukken bevat een verzameling van good practices die uit de praktijk zijn verzameld. In de beschrijving van elke good practice wordt steeds de volgende structuur gebruikt:

- **Naam in de paragraaftitel:** een herkenbare naam waaraan de inhoud van de good practice te herkennen is.
- **Vraagstelling:** een beschrijving van de vraag die aanleiding is voor het expliciet maken van deze good practice.
- **Oplossing:** een beschrijving en visualisatie van de oplossing die de voorkeur verdient om de vraag te kunnen beantwoorden.
- **Gevolgen:** een beschrijving van de gevolgen van de oplossing voor bijvoorbeeld de communicatie met belanghebbenden, de gevolgen voor de implementatie, e.d.
- **Alternatieven:** een beschrijving van alternatieve oplossingen met de afwegingen om hiervoor te kiezen.
- **Relaties met andere good practices:** een beschrijving van de relaties die er zijn met andere good practices.

In dit document zijn twee soorten good practices verzameld:

- Good practices die antwoord geven op ArchiMate-concept vragen (hoe gebruik je een bepaald concept in de praktijk?).



- Good practices die antwoord geven op modelleervragen.

De good practices zijn op de volgende wijze geordend:

- Modelleren in de bedrijfslaag: hoofdstuk 2.
- Modelleren in de applicatielaag: hoofdstuk 3.
- Modelleren in de infrastructuur-/technologielaag: hoofdstuk 4.
- Modelleren over de grenzen van lagen heen: hoofdstuk 5.

#### **1.4 Wijzigingen ten opzichte van versie 3.0**

Dit document is een vernieuwde versie van het boekje 'ArchiMate in de praktijk' waarvan een derde versie in 2009, door de werkgroep Gebruik/Tools binnen het ArchiMate Forum, is uitgebracht. In deze versie 4.0 zijn nieuwe good practices toegevoegd. Tevens zijn de bestaande teksten verbeterd en zijn de figuren qua stijl gelijk getrokken.

De nieuwe good practices zijn:

- Informatiedomein.
- 'Nesten' en impliciet afgeleide relaties.
- File transfer.

#### **1.5 Website**

Meer informatie over het ArchiMate Forum, de activiteiten van de werkgroep Gebruik/Tools en de verzamelde good practices zijn te vinden via de websites:

[www.opengroup.org/archimate](http://www.opengroup.org/archimate) en [www.archimate.nl](http://www.archimate.nl). Op de tweede website worden tevens nieuwe good practices gepubliceerd alvorens deze in een volgende versie van het boekje worden opgenomen.

#### **1.6 Oproep voor nieuwe good practices**

De inhoud van dit document is zeker niet volledig en laat nog vragen onbeantwoord. Dit document is bedoeld als groeidocument. Het bevat een verzameling van good practices die uit de praktijk zijn verzameld.

We hopen dat de lezer hiermee in de praktijk aan de slag kan, en dat vervolgens op basis van dit gebruik nieuwe inzichten ontstaan die vervolgens in dit document verwerkt zullen worden.

Heb je modelleervragen waar je graag antwoord op zou willen krijgen? Heb je zelf ook good practices die je wil delen met vakgenoten? Of wil je ook deelnemen aan de werkgroep Gebruik/Tools binnen het ArchiMate Forum, waar we de good practices verzamelen, bespreken en verwerken in dit document? Meld dit dan aan de voorzitters van de werkgroep via de mailadressen: [h.vandenberg@bizzdesign.nl](mailto:h.vandenberg@bizzdesign.nl) en [egon.willemsz@uwv.nl](mailto:egon.willemsz@uwv.nl). Of telefonisch bij Harmen van den Berg (06 5119 8282) of Egon Willemsz (06 5235 3089).



## 2 Modelleren in de bedrijfslaag

### 2.1 Overzicht good practices

In dit hoofdstuk zijn de onderstaande good practices opgenomen:

1. Bedrijfsservice, bedrijfsfunctie en bedrijfsproces.
2. Afbakening bedrijfsservice.
3. Afbakening bedrijfsfunctie.
4. Afbakening bedrijfsproces.
5. Bedrijfsproces en bedrijfsinteractie.
6. Viewpoints voor procesmodellering.
7. Samenwerking tussen organisaties.
8. Informatiedomein.

### 2.2 Bedrijfsservice, bedrijfsfunctie en bedrijfsproces

**Vraagstelling:** In de bedrijfslaag worden ondermeer de concepten bedrijfsservice, bedrijfsfunctie en bedrijfsproces gehanteerd. Het precieze onderscheid tussen deze concepten is niet altijd even duidelijk, dat wil zeggen dat bij toepassing in de praktijk er vaak verwarring ontstaat of iets nu een bedrijfsservice, een bedrijfsfunctie of een bedrijfsproces is. Hieronder geven we definities en omschrijvingen van deze concepten.

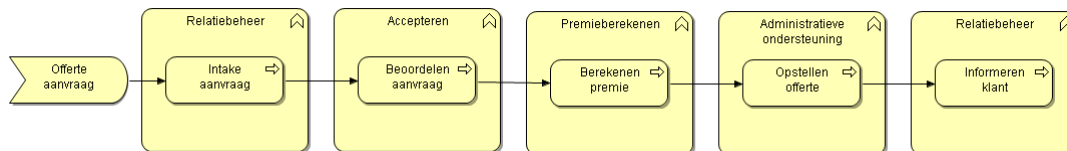
**Oplossing:** Een bedrijfsservice is de bijdrage die vanuit de organisatie wordt geleverd aan de omgeving. Er kan onderscheid worden gemaakt tussen interne en externe services. Interne services zijn de bijdragen die worden geleverd binnen het domein waar de service onderdeel van uitmaakt. Externe services zijn de bijdragen die beschikbaar worden gesteld aan andere domeinen of aan de omgeving (bijvoorbeeld klanten).

Een bedrijfsfunctie is een aandachtsgebied waaraan het bedrijf structureel aandacht wil besteden (= energie in wil stoppen, structureel middelen voor wil inzetten) om zijn bedrijfsdoelstelling te realiseren. Een bedrijfsfunctie kan daarom ook gezien worden als een groepering van intern gedrag op basis van een bepaald criterium (bijvoorbeeld plaats (dezelfde afdeling), communicatie, benodigde competenties, gedeelde bronnen en gedeelde kennis). Een bedrijfsfunctie representeert een stuk toegevoegde waarde van de organisatie.

Een bedrijfsproces is een eenheid van intern gedrag of een verzameling van causaal (volgorde, afhankelijkheid) gerelateerde eenheden van intern gedrag, met als doel een voorgedefinieerde verzameling van producten en diensten te produceren. Een bedrijfsproces kan bestaan uit deelprocessen of activiteiten. Een bedrijfsproces wordt getriggerd (opgestart) door één of meerdere business events (gebeurtenissen) of andere bedrijfsprocessen.

Informeel zou dus gezegd kunnen worden dat een bedrijfsproces bestaat uit een aantal activiteiten of deelprocessen die in een bepaalde volgorde worden uitgevoerd. Elke activiteit maakt onderdeel uit, of hoort bij, een bedrijfsfunctie. Met andere woorden, een proces rijgt

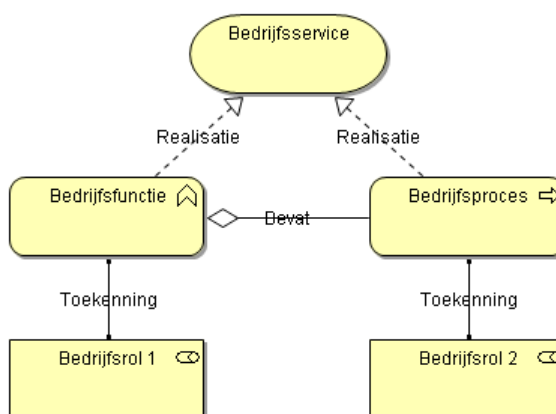
een keten van activiteiten aan elkaar die elk horen bij bedrijfsfuncties, zoals ook gevisualiseerd in onderstaand schema. Overigens is het niet zo dat één proces slechts bij één bedrijfsfunctie hoort (zoals in dit voorbeeld): een bedrijfsfunctie omvat vrijwel altijd meerdere activiteiten en processtappen, en een proces zal vaak door meerdere bedrijfsfuncties worden gerealiseerd.



Figuur 2-1: Voorbeeld van bedrijfsfuncties in relatie tot processen

Zowel bedrijfsfuncties als processen beschrijven de interne gang van zaken binnen een aandachtsgebied of organisatie. Een bedrijfsservice beschrijft juist datgene dat van buitenaf zichtbaar en bruikbaar is van processen en functies. Het beschrijft de dienst die gebruikt kan worden, zonder dat duidelijk is hoe die dienst met behulp van processen of bedrijfsfuncties gerealiseerd wordt.

In onderstaande figuur zijn de belangrijkste relaties tussen deze concepten aangegeven (een deel van het metamodel):



Figuur 2-2: Belangrijkste relaties tussen concepten

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Een soortgelijk onderscheid als tussen bedrijfsservice en bedrijfsproces / bedrijfsfunctie geldt tussen applicatieservice en applicatiefunctie. Soortgelijke criteria en heuristieken kunnen daarom ook op de applicatielaag gebruikt worden.

## 2.3 Afbakening bedrijfsservice

**Vraagstelling:** In de bedrijfslaag wordt ondermeer het concept bedrijfsservice gehanteerd. Maar hoe onderken je een bedrijfsservice, en hoe baken je deze af?

**Oplossing:** Als iets als een bedrijfsservice wordt gedefinieerd moet de service in principe door meerdere gebruikers af te nemen zijn. Services kunnen gegroepeerd worden met behulp van aggregatierelaties. Dit betekent dat het alleen zinvol is om een service te decompونeren (via aggregatie) in deelservices als deze deelservices ook zelfstandig als services af te nemen zijn.

Hanteer de volgende heuristieken om bedrijfsservices te onderkennen:

- Onderken een service vanuit het perspectief van de serviceafnemer. De service moet herkenbaar en bruikbaar zijn voor de afnemer. De naamgeving van services moet ook vanuit het perspectief van de serviceafnemer gebeuren.
- Onderken services op basis van de activiteiten die onderkend worden in de bedrijfslaag, en op basis van de producten die geleverd worden.
- Onderken verschillende services voor verschillende belangen (concerns).
- Vermijd overlap in geboden services: verschillende services bieden verschillend gedrag. Overlap in gedrag is een indicator om het overlappende gedrag als aparte service te beschouwen.
- Een service wordt gerealiseerd door één of meer bedrijfsfuncties of processen, die concreet intern gedrag van de organisatie representeren. Een bedrijfsfunctie mag meerdere bedrijfsservices realiseren.
- Modelleer van een bedrijfsservice altijd door welke bedrijfsfuncties of processen deze gerealiseerd wordt en door welke bedrijfsprocessen (in het geval van interne services) deze gebruikt wordt.
- Een interne bedrijfsservice wordt altijd gebruikt door minimaal één bedrijfsfunctie of bedrijfsproces.
- Houd services consistent: zorg dat soortgelijk gedrag op een soortgelijke manier in services aangeboden wordt.
- Gebruik services om implementatiedetails af te schermen. Het moet voor een serviceafnemer voldoende zijn om te weten *dat* een bepaalde service aangeboden wordt en hoe de afnemer hiervan gebruik moet maken. Een serviceafnemer hoeft niet te weten *hoe* een service gerealiseerd wordt.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Soortgelijke criteria en heuristieken kunnen ook op de applicatielaag voor applicatieservices en op de infrastructuurlaag voor infrastructurele services gebruikt worden.

## 2.4 Afbakening bedrijfsproces

**Vraagstelling:** In de bedrijfslaag wordt ondermeer het concept bedrijfsproces gehanteerd. Maar hoe onderken je een bedrijfsproces, en hoe baken je deze af?

**Oplossing:** Hanteer de volgende heuristieken om bedrijfsprocessen te onderkennen:

- Een bedrijfsproces wordt getriggerd door een bedrijfsevent, en levert uiteindelijk een dienst of product op voor een klant, of deelproducten of deeldiensten die gebruikt worden als onderdeel voor een dienst of product voor een klant.
- Verdeel een proces in fasen die achtereenvolgens doorlopen worden. Voorbeelden van fasen zijn aanvraag, afhandeling en after sales. Zo'n fase komt vaak overeen met een werkproces of functie binnen de organisatie. Een veel voorkomende vorm van fasering is de waardeketen (value chain). U herkent fasen vaak aan de toewijzing van acties aan verschillende actoren, gecombineerd met een tijdsvolgorde tussen die acties.
- Groepeer acties op grond van het tijdstip waarop zij plaatsvinden (online- of batchverwerking, overdag of 's nachts).
- Deel een proces op in stukken op grond van de kennis en vaardigheden die zijn vereist om bepaalde acties uit te voeren. Dit blijkt onder meer uit de benodigde functies die zijn aangegeven bij elke taak.
- Deel een proces op in stukken op grond van functiescheiding of andere vormen van interne controlemaatregelen.
- Deel een proces in stukken op grond van het geografisch gebied (fysieke locatie) waar de activiteiten plaatsvinden, bijvoorbeeld naar regio.
- Verdeel een proces in deelprocessen die (grotendeels) onafhankelijk uitgevoerd kunnen worden. Het gaat hierbij vaak om verschillende (tussen)producten die worden opgeleverd.
- Baken expliciet de deelprocessen af die vaker voorkomen. Deze algemene deelprocessen kunt u vervolgens apart definiëren, en dan meermaals gebruiken.
- Baken expliciet de deelprocessen af die als geheel herhaald worden.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Niet van toepassing.

## 2.5 Afbakening bedrijfsfunctie

**Vraagstelling:** In de bedrijfslaag wordt ondermeer het concept bedrijfsfunctie gehanteerd. Maar hoe onderken je een bedrijfsfunctie, en hoe baken je deze af?

**Oplossing:** Hanteer de volgende heuristieken om bedrijfsfuncties te onderkennen:

- Maak een aparte functie voor de *interacties met elke omgevingsactor*. Bijvoorbeeld een functie voor toeleveranciers, afnemers en de overheid. Deze heuristiek kan worden verbijzonderd naar:

- *Scheiden relatiebeheer van overige functies*: Maak een aparte functie voor de activiteiten die direct te maken hebben met het contact met de klant. In principe gaat het hier om de externe klant van het bedrijf, maar voor grotere bedrijven kan het ook toepasbaar zijn voor een interne klant (bijvoorbeeld een andere afdeling).
- *Scheiding naar markt*: Maak een aparte functie voor elke klantgroep/doelgroep van het bedrijf. Voorbeelden van dergelijke groepen zijn zakelijke klanten en particulieren.
- Maak aparte functies voor processen met verschillende soorten *triggers* (business events). Zo kunnen processen worden geïdentificeerd met periodieke triggers (maandelijks ontvangen van declaraties) en met incidentele triggers (vragen om offerte). Een verbijzondering hiervan is:
  - *Gevalsonderscheid*: Maak aparte functies voor activiteiten die betrekking hebben op verschillende 'gevallen'. Zo kan een bedrijf onderscheid maken tussen verschillende schadegevallen: schades kleiner respectievelijk groter dan f 1000,-. In dit geval zou er dus een functie voor kleine en een functie voor grote schadegevallen onderscheiden kunnen worden.
- Maak een aparte functie voor elke *productgroep*, bijvoorbeeld voor activiteiten rond schadeverzekeringen en rond pensioenverzekeringen. Algemener kan een onderscheid worden gemaakt rondom een willekeurig bedrijfsobject:
  - *Scheiding naar bedrijfsobject*: Maak aparte functies voor een groep activiteiten, indien ze werken op een bepaald bedrijfsobject. Zo kunnen er functies zijn die betrekking hebben op facturen, op geldstromen, op offertes, op klantdossiers, etc.
- Maak een aparte functie voor elke fase of toestand waarin een product zich kan bevinden. In geval van een schadeverzekering zouden dit bijvoorbeeld aparte functies kunnen opleveren voor aanvragen, premieheffing en afhandeling van schadeclaims, beëindiging en beheer.
- Maak een aparte functie voor een groep activiteiten, indien deze activiteiten speciale 1) vaardigheden, 2) expertise of 3) verantwoordelijkheden vereisen. Voorbeelden hiervan zijn juridische kennis, actuariële expertise, communicatieve vaardigheden of autorisaties voor het nemen van bepaalde beslissingen (fiattering).
- Maak aparte functies voor activiteiten die het primaire proces *besturen*. Het gaat daarbij in het bijzonder om planning en controle.
- Maak aparte functies voor activiteiten die *veranderingen* aanbrengen aan het primaire proces of de implementatie daarvan. Dit leidt tot aparte functies voor bijvoorbeeld marketing, productontwikkeling en systeemontwikkeling.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

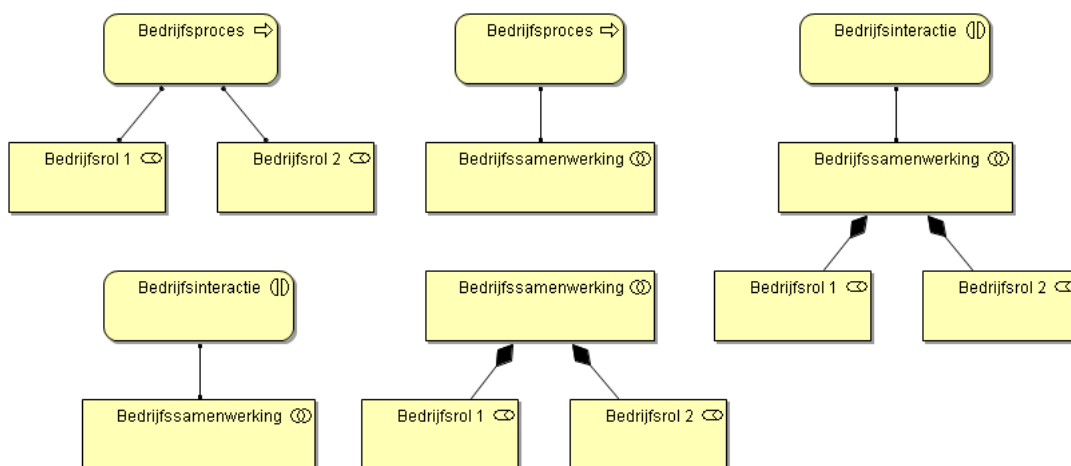
**Relaties met andere good practices:** Soortgelijke criteria en heuristieken kunnen ook op de applicatielaag voor applicatiefuncties gebruikt worden.

## 2.6 Bedrijfsproces en bedrijfsinteractie

**Vraagstelling:** ArchiMate kent naast het concept bedrijfsproces ook het concept bedrijfsinteractie. Een bedrijfsinteractie is een proces dat door twee of meer actoren (een collaboratie of samenwerking van actoren) wordt uitgevoerd. Waarom is het concept bedrijfsinteractie nodig naast het concept bedrijfsproces? Is het niet voldoende om een

proces te koppelen aan een collaboratie (samenwerking) van actoren, en daarmee uit te drukken dat het proces eigenlijk een interactie is?

**Oplossing:** Een bedrijfsinteractie is gedefinieerd als een specialisatie van bedrijfsproces (of preciezer van het generieke concept gedrag). Door een bedrijfsinteractie te gebruiken voeg je dus informatie toe, namelijk dat het gedrag door een samenwerking van rollen of actoren wordt uitgevoerd. Door een bedrijfsproces te koppelen aan een bedrijfssamenwerking geef je dezelfde informatie weer, namelijk dat het gedrag door een samenwerking van rollen of actoren wordt uitgevoerd. In bijgaande figuur zijn enkele varianten afgebeeld:



Figuur 2-3: Varianten van bedrijfsproces en bedrijfsinteractie

Linksboven is gemodelleerd dat een bedrijfsproces is toegekend aan twee rollen, middenboven dat een bedrijfsproces is toegekend aan een bedrijfssamenwerking, linksonder dat een interactie is toegekend aan een bedrijfssamenwerking, middenonder dat een bedrijfssamenwerking bestaat uit twee rollen, en geheel rechts de combinatie dat een interactie is toegekend aan een bedrijfssamenwerking die weer bestaat uit twee rollen.

Deze redundantie heeft echter wel een goede reden. Soms wil je niet de bedrijfssamenwerking expliciet tekenen (bijvoorbeeld als de nadruk ligt op het gedrag, en niet op de actoren), maar toch aangeven dat het om een interactie door meerdere rollen gaat. Een ander voorbeeld is dat je wel wilt aangegeven dat het een samenwerking is, maar niet door welke partijen of actoren de samenwerking wordt vormgegeven; dit kan bijvoorbeeld zinvol zijn in ontwerptraject waar nog niet bepaald is welke partijen uiteindelijk zullen bijdragen aan een proces of dienst.

**Gevolgen:** Het doel dat een architect heeft met het modelleren of visualiseren van een bedrijfsinteractie of -samenwerking bepaalt dus in grote mate welke oplossing hiervoor gekozen wordt en welk aspect de nadruk krijgt.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Een soortgelijk onderscheid als hiervoor beschreven speelt ook op de applicatielaag, namelijk rond applicatie-interactie en applicatiesamenwerking. De hiervoor beschreven voorbeelden en richtlijnen zijn ook hierop van toepassing.



## 2.7 Viewpoints voor procesmodellering

**Vraagstelling:** Hoe modelleer je in ArchiMate een proces? Het ArchiMate boek geeft een aantal viewpoints, maar welke combinatie van viewpoints heb je nodig om een proces in opeenvolgende stappen te beschrijven?

**Oplossing:** Het ArchiMate boek geeft twee viewpoints voor procesmodellering: het Bedrijfsprocessamenwerkings viewpoint en het Bedrijfsproces viewpoint. Het verschil tussen deze viewpoints is dat in het Bedrijfsprocessamenwerkings viewpoint de relatie *tussen* processen en de relatie met de omgeving wordt weergegeven, terwijl in het Bedrijfsproces viewpoint één bedrijfsproces wordt uitgelicht. We concentreren ons hier op het Bedrijfsproces viewpoint; het Bedrijfsprocessamenwerkings viewpoint is analoog.

Bij het modelleren van een proces in ArchiMate kunnen de volgende aspecten worden weergegeven: de onderdelen van het proces, hun samenhang, de toewijzing van procesdelen aan rollen, actoren of bedrijfsonderdelen en de ondersteuning door applicaties. Je kunt dit doen in de volgende stappen:

1. *Modelleer het bedrijfsproces op hoofdlijnen:* Decomponeer het bedrijfsproces en leg verbanden door bedrijfsevents en triggeringrelaties. Daaraan toegevoegd kan worden de wijze waarop het proces bedrijfsservices verleent aan zijn omgeving.
2. *Informatie-overdracht:* Modelleer de informatie-overdracht tussen processen door het toevoegen van flowrelaties tussen processen, of het lezen en schrijven van bedrijfsobjecten.
3. *Toewijzing aan organisatie:* Modelleer welk organisatie-onderdeel het bedrijfsproces uitvoert. Dit kan gevisualiseerd worden door toekenningsrelaties of door gebruik te maken van procesbanen.
4. *Ondersteuning door applicaties:* Modelleer de wijze waarop applicaties de bedrijfsprocessen ondersteunen. Hiervoor kan het Applicatiegebruik viewpoint worden gebruikt.

Eventueel kan nog een algemeen onderscheid gemaakt worden tussen een 'logische' en een 'concreet' niveau van modelleren. Een *logisch model* geeft de functionele structuur en de logische samenhang van bedrijfsprocessen weer. Er wordt niet ingegaan op tijd, plaats, mensen en machines. Een *inrichtingsmodel* komt sterk overeen met wat je in de werkelijkheid ziet of wilt realiseren. Het beschrijft *fysieke* zaken zoals mensen en machines, met daaraan toegewezen concrete taken. Indien een organisatie een standaard typering van processen kent / wil onderkennen, moet hiervoor een specifieke afspraak worden gemaakt bij de opzet van het modelleren met ArchiMate hiervoor. De relatie tussen 'logisch proces' en 'fysiek proces' is van het type compositie.

Tenslotte kun je een proces specialiseren in een aantal typen processen. Bijvoorbeeld kan een proces 'Aanvragen verzekering' gespecialiseerd worden in een drietal subprocessen 'Aanvragen reisverzekering' en 'Aanvragen schadeverzekering'.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Zie ook de good practices 'Bedrijfsservice, bedrijfsfunctie en bedrijfsproces' en 'Afbakening bedrijfsproces'.

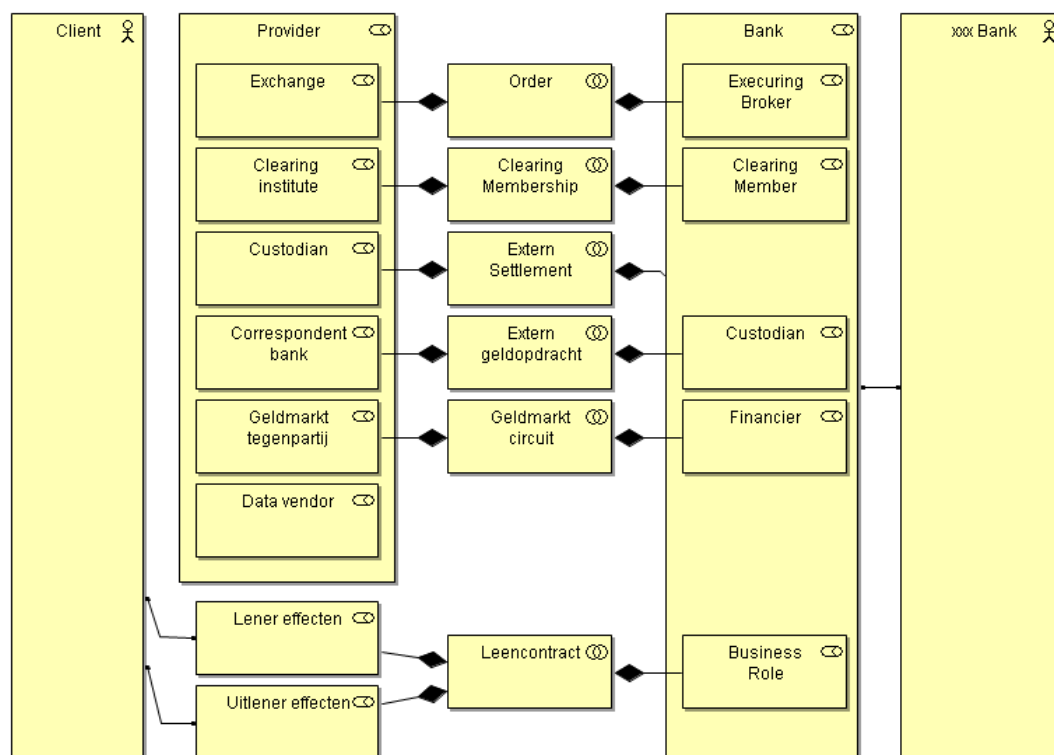
## 2.8 Samenwerking tussen organisaties

**Vraagstelling:** Hoe modelleer je de samenwerking tussen organisaties? Wat wordt bedoeld met bedrijfssamenwerking en hoe verhouden die zich de concepten bedrijfsinteractie en bedrijfsrol over en weer?

**Oplossing:** Voor een goed inzicht in de samenwerking tussen organisaties is de samenwerkingsview zeer behulpzaam. Deze view laat zien hoe de diverse rollen en actoren samenwerken in een keten. Enkele concepten die hierbij van belang zijn:

- *Bedrijfssamenwerking:* Een, mogelijk tijdelijk, collectief van rollen die samenwerking vertonen (= interactie hebben). Het concept bedrijfsinteractie is het gedrag wat getoond wordt in de samenwerking van twee of meer bedrijfsrollen.
- *Bedrijfsactor:* Een actieve entiteit die gedrag vertoont. Het kan een persoon zijn, bv een klant, maar ook een groep mensen, bv een bedrijfsonderdeel.
- *Bedrijfsrol:* Het gedrag uitgevoerd door een actor die de rol vervult

Het onderstaande voorbeeld geeft een goed inzicht in de rollen in de keten en de samenwerking.



Figuur 2-4: Samenwerking tussen organisaties

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

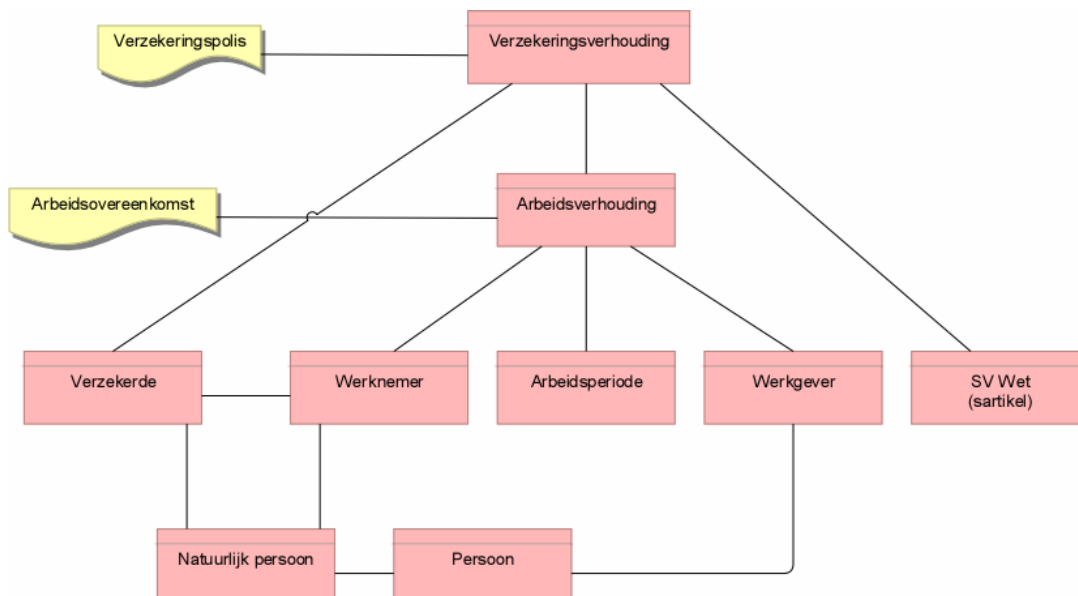
**Relaties met andere good practices:** Zie ook de good practices ‘Bedrijfsservice, bedrijfsfunctie en bedrijfsproces’ en ‘Bedrijfsproces en bedrijfsinteractie’.

## 2.9 Informatiedomein

**Vraagstelling:** In onze organisatie is het van belang een eenduidig begrippenkader te hebben. Hiervoor wil ik een informatiemodel (objectmodel) gebruiken. Hoe kan ik dat met ArchiMate modelleren en welke soorten relaties tussen de objecten kan ik het best gebruiken?

**Oplossing:** Aan de hand van onderstaand fictief voorbeeld wordt een en ander beschreven. Bedrijfsregels: ‘Een Arbeidsverhouding geeft het verband weer tussen een Werknemer en een Werkgever. Op grond van de Arbeidsverhouding is de Werknemer verplicht verzekerd in het kader van een bepaalde SV-wet. Daarnaast kan een niet Werknemer zich vrijwillig verzekeren. Binnen een Arbeidsverhouding zijn er Arbeidsperioden die zich onderscheiden door gelijkblijvende eigenschappen (bijv. loongegevens)’

Een eenvoudig informatiemodel op basis van bovenstaande beschrijving ziet er als volgt uit:



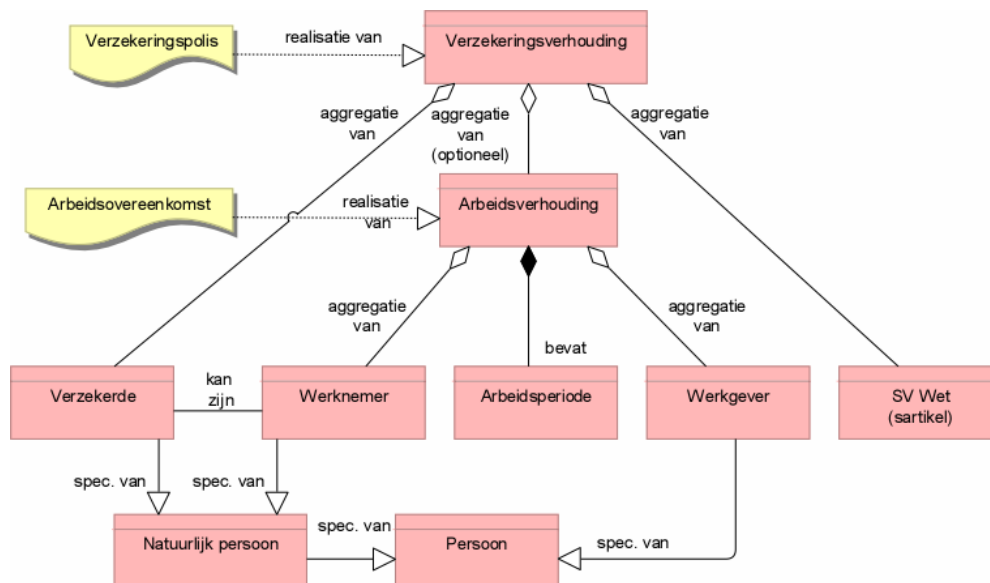
Figuur 2-5: Eenvoudig informatiemodel

De relaties tussen bedrijfsobjecten zijn in deze variant vormgegeven door associaties, wat alleen wil zeggen dat er tussen deze bedrijfsobjecten een verband bestaat. De associatie is de zwakste vorm van een relatietype in ArchiMate. Het bovenstaande voorbeeld is dan ook semantisch zwak; er bestaan associaties tussen de bedrijfsobjecten maar wat die betekenen wordt niet uitgedrukt.

Een model als bovenstaand is bij uitstek geschikt voor communicatie met betrokkenen die weinig of geen kennis van gegevensmodellering hebben, zoals stakeholders vanuit de business of vanuit management, of in een fase waarin het informatiemodel wordt ontwikkeld (en de precieze relaties tussen de informatieobjecten nog niet bekend is).

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Een meer gedetailleerde variant ziet er als volgt uit:



Figuur 2-6: Gedetailleerd informatiemodel

De relaties tussen bedrijfsobjecten zijn hier vormgegeven door sterkere vormen van relatietypen. Hierdoor krijgt het model meer betekenis. In het voorbeeld worden de volgende relatietypen gebruikt:

- *Aggregatie*: De Arbeidsverhouding is een aggregatie van een Werknemer en een Werkgever.
- *Compositie*: De Arbeidsverhouding bevat Arbeidsperioden.
- *Specialisatie*: Een Werknemer is een specialisatie van een Natuurlijk persoon evenals de Verzekerde.
- *Realisatie*: De Arbeidsovereenkomst is een realisatie van een Arbeidsverhouding.

In ArchiMate is het niet standaard mogelijk om aan deze relaties optionaliteiten en cardinaliteiten toe te kennen. Hiervoor zijn de specifieke domeintalen (zoals UML) bedoeld. Wel kunnen in sommige tools dergelijke eigenschappen worden toegevoegd.

In situaties waarin de doelgroep bekend is met gegevensmodellering, en/of wanneer er behoefte is aan een meer preciezere vastlegging van de relaties tussen informatieobjecten, is deze tweede variant beter geschikt dan de eerste variant.

Ook is het uiteraard mogelijk om (in verband met communicatiedoeleinden) de eerste variant af te leiden uit de tweede; daarbij wordt de vereenvoudigde weergave getoond, terwijl het meer gedetailleerde model onderliggend is.

**Relaties met andere good practices:** Niet van toepassing.

## 3 Modelleren in de applicatielaag

### 3.1 Overzicht good practices

In dit hoofdstuk zijn de onderstaande good practices opgenomen:

1. Applicatielandschap.
2. Koppeling tussen applicaties.
3. Functionaliteit van een applicatie.
4. Websites als applicatiecomponent en het gebruik van services.

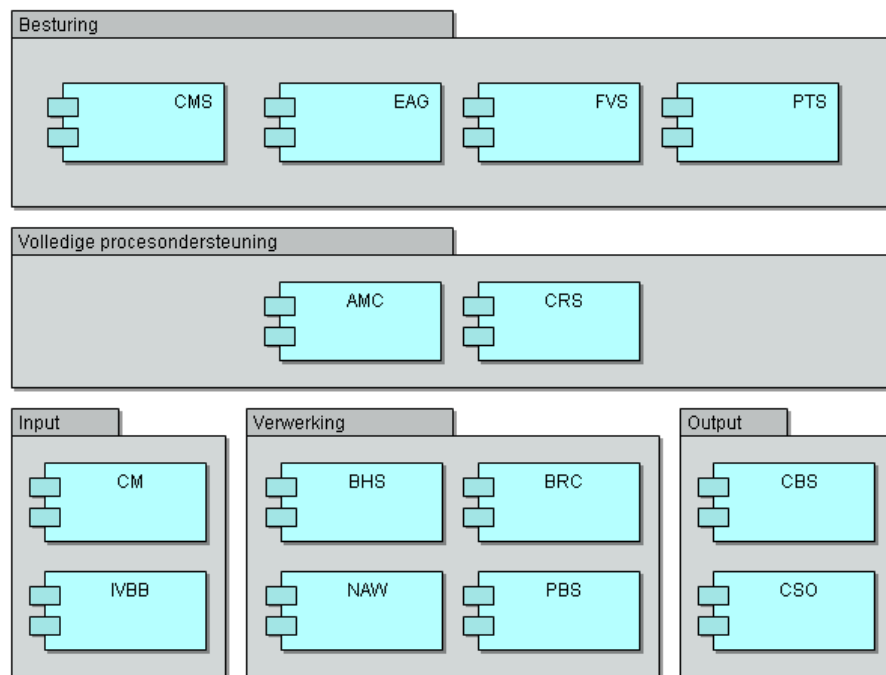
### 3.2 Applicatielandschap

**Vraagstelling:** Mijn applicatielandschap is zeer complex, hoe krijg ik daar grip op, hoe modelleer ik dit?

Vaak wordt een overzicht van applicaties getoond in een applicatielandschap. Dit om een overzicht te verschaffen van de relaties tussen de applicaties. Vaak geeft dit een groot aantal verbindingslijnen en eigenlijk geen enkel overzicht. Dergelijke overzichten geven alleen aan hoe complex het applicatielandschap eruit ziet. Het geeft geen houvast voor het onderhouden van je applicatieportfolio.

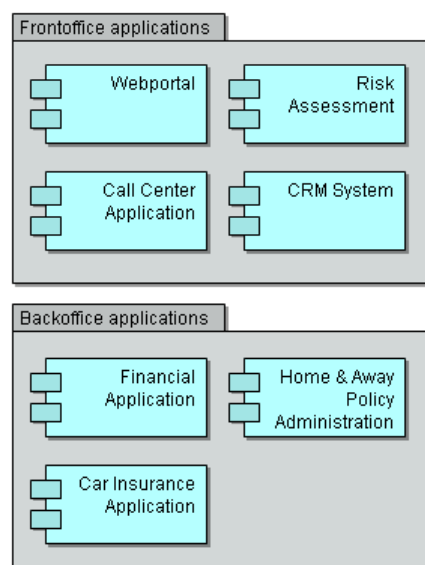
**Oplossing:** Het is belangrijk om de overzichten leesbaar te houden. Houd een regel aan van maximaal 10 a 15 applicaties per overzicht. Zijn het er meer, dan is het aan te bevelen om gebruik te maken van een algemeen model met daarin de hoofdgroepering plus een detailmodel per groep. Kies een viewpoint die voor de doelgroep van belang is. De vorm is altijd een Applicatiestructuur viewpoint. Een classificatie kan worden gedaan via proces, organisatie, gegevens, type functionaliteit of infrastructuur. Groepeer de applicaties per gebied en verbind alleen de gebieden. Bij grotere aantallen is het tonen van relaties via een matrix veel overzichtelijker.

Hieronder is een voorbeeld opgenomen van een applicatielandschap ingedeeld naar de type functionaliteit van de applicaties (besturing, inputverwerking, verwerking, outputverstrekking en volledige procesondersteuning).



Figuur 3-1: Voorbeeld van applicatielandschap ingedeeld naar type functionaliteit van applicaties

Hieronder is een voorbeeld opgenomen van een applicatielandschap ingedeeld naar het gebruik door de frontoffice en backoffice.



Figuur 3-2: Voorbeeld van applicatielandschap ingedeeld naar gebruik door frontoffice en backoffice

**Gevolgen:** Voor een overzicht van alle relaties zijn meerdere diagrammen nodig. Gebruik hiervoor een matrix. Een tool kan in dit geval zeer waardevol zijn om deze informatie vast te leggen en bijvoorbeeld een kruistabel mee te maken.

**Alternatieven:** Een alternatief is de grote 'plaat' met alle applicaties en de vele lijnen, of een (overzichtelijke) matrix of tabel. Elk van dergelijke overzichten kan zeker toegevoegde

waarde hebben, bijvoorbeeld om totaaloverzichten te geven of veranderingen zichtbaar te maken over het totale landschap.

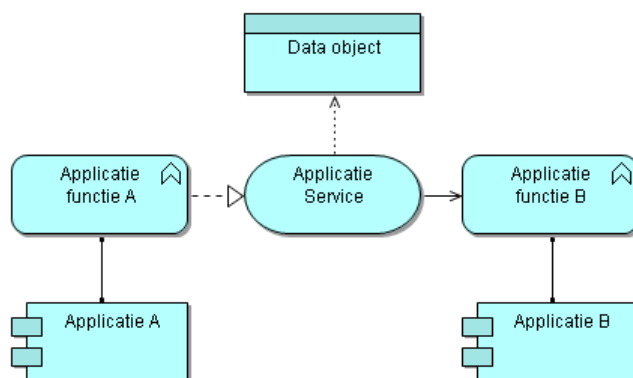
**Relaties met andere good practices:** Hier ligt een belangrijke relatie met de good practice 'Koppeling tussen applicaties'.

### 3.3 Koppeling tussen applicaties

**Vraagstelling:** Het is wenselijk dat koppelingen tussen applicaties in beeld worden gebracht. Welke concepten binnen ArchiMate kunnen hiervoor worden gebruikt?

**Oplossing:** Een applicatie wordt gemodelleerd als applicatiecomponent met bijbehorende applicatiefuncties. Een koppeling tussen twee applicaties wordt gemodelleerd via een applicatieservice. De applicatieservice levert gegevens door middel van een toegangsrelatie met een data-object.

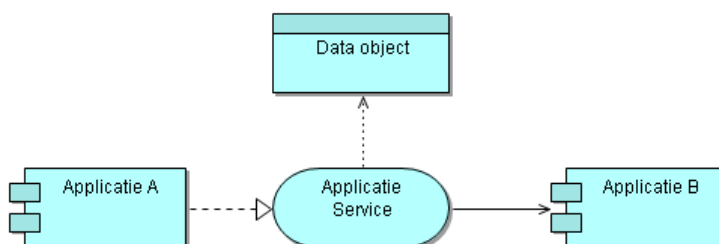
In het onderstaande voorbeeld is gemodelleerd dat applicatiefunctie A een applicatieservice realiseert die door B wordt gebruikt. De applicatieservice levert de gegevens die in het data-object zijn opgenomen.



Figuur 3-3: Applicatiekoppeling middels applicatiefunctie, applicatieservice en gekoppeld data-object

**Gevolgen:** Als via deze oplossing de huidige koppelingen tussen applicaties binnen een bedrijf in beeld worden gebracht, dan kan de suggestie worden gewekt dat er al sprake is van een servicegeoriënteerde architectuur terwijl daar in werkelijkheid nog geen sprake van is.

**Alternatieven:** Met inachtneming van de compositieregels (zie good practice 'Vereenvoudigingen met behoud van consistentie'), kunnen de applicatiefuncties in het model weggelaten worden. Onderstaande figuur toont het resultaat.



Figuur 3-4: Applicatiekoppeling middels applicatieservice en gekoppeld data-object

Ook is het mogelijk bovenstaande koppeling aan te geven zonder dat er een data-object in meegenomen wordt:



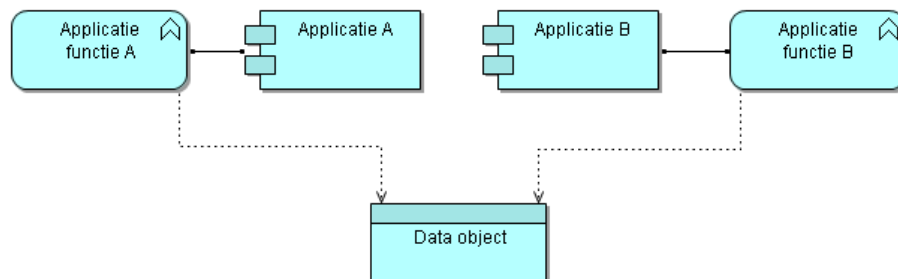
Figuur 3-5: Applicatiekoppeling middels applicatieservice

Niet elke organisatie heeft al een servicegeoriënteerde architectuur. Het is dan ook mogelijk om de koppeling van applicaties te modelleren met behulp van een flow-relatie tussen de applicaties. Met gebruikmaking van een afgeleide relatie ziet dat er uit als hieronder aangegeven. Applicatie A levert klantgegevens aan applicatie B.



Figuur 3-6: Applicatiekoppeling middels flowrelatie

In sommige gevallen vindt de koppeling van de applicaties niet rechtstreeks plaats, maar gebeurt dat via een database. Daarbij schrijft de ene applicatie in een tabel, die door een andere applicatie gelezen wordt. Onderstaande figuur geeft dit weer. Applicatie A schrijft, Applicatie B leest datzelfde data-object.



Figuur 3-7: Applicatiekoppeling middels data-object

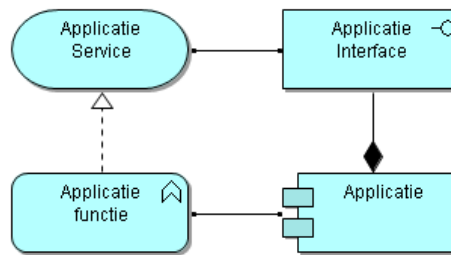
**Relaties met andere good practices:** Dit kan worden gebruikt bij het opbouwen van een applicatielandschap via de good practice 'Applicatielandschap'. Dit is ook van toepassing bij de good practice 'Servicebroker'.

### 3.4 Functionaliteit van een applicatie

**Vraagstelling:** Hoe breng je met ArchiMate de extern zichtbare functionaliteit van applicaties in beeld?

**Oplossing:** Het ArchiMate-concept dat expliciet bedoeld is voor het modelleren van extern zichtbare functionaliteit is de applicatieservice die via een realisatierelatie is gekoppeld aan een applicatiefunctie die hoort bij een applicatie. Daarbij kan naar keuze de bijbehorende interface(s) worden weergegeven. De onderstaande figuur geeft deze modellering weer.



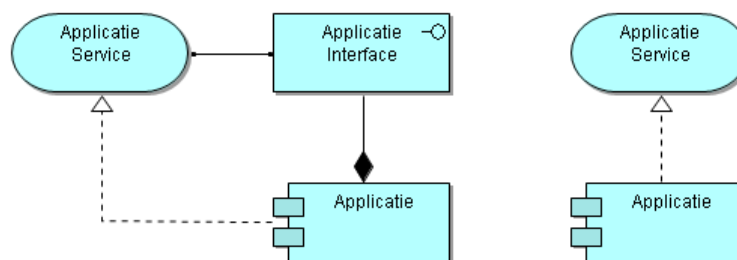


Figuur 3-8: Extern zichtbare functionaliteit

Deze manier van modelleren is zowel te gebruiken voor interne als externe applicatieservices.

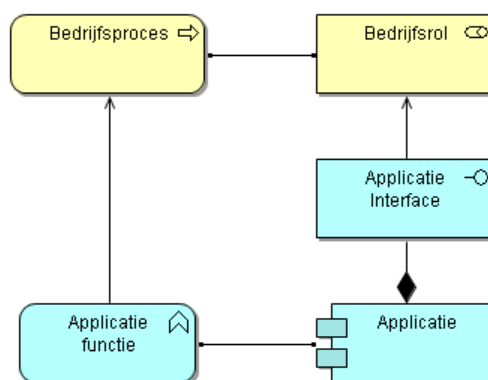
**Gevolgen:** Als een applicatie veel functionaliteit levert, dan kan een dergelijk model onoverzichtelijk worden door de veelheid aan realisatielaties. Tevens geldt ook hier dat de suggestie kan worden gewekt dat er al sprake is van een servicegeoriënteerde architectuur terwijl daar in werkelijkheid nog geen sprake van is.

**Alternatieven:** Deze modellering kan vereenvoudigd worden door de applicatiefunctie weg te laten en de realisatielatie af te laten leiden naar de applicatiecomponent (onderstaande figuur links). Tevens kan, indien deze voor het doel van het model niet relevant is, de interface weggelaten worden (onderstaande figuur rechts).



Figuur 3-9: Extern zichtbare functionaliteit - vereenvoudigd

Wanneer er geen services onderscheiden worden, kan de extern zichtbare functionaliteit van een applicatie weergegeven worden door de applicatiefunctie via een interface te laten ontsluiten aan de 'gebruiker'. In het onderstaande voorbeeld wordt een Bedrijfsproces ondersteund door een Applicatiefunctie, die ontsloten wordt door een Applicatie-interface.



Figuur 3-10: Extern zichtbare functionaliteit gekoppeld aan een procesrol

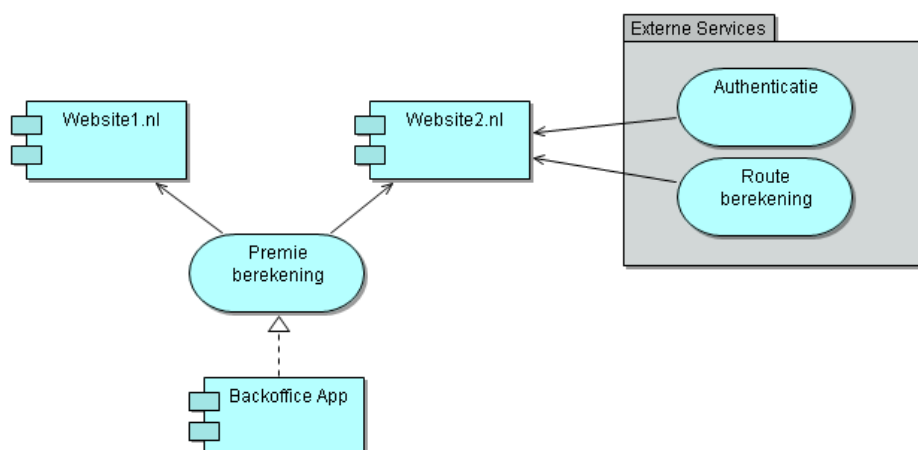
In de laatste view is het ook mogelijk om de applicatie-interface weg te laten. Er bestaat dan een gebruik door relatie tussen de applicatie en de bedrijfsrol.

**Relaties met andere good practices:** Voor niet interactieve functionaliteit ligt een relatie met de good practice 'Koppeling tussen applicaties'. Het in beeld brengen van functionaliteit van een applicatie lijkt veel op het in beeld brengen van de applicaties binnen een applicatieportfolio. De good practice 'Applicatielandschap' is ook toepasbaar voor de functionaliteit van een applicatie.

### 3.5 Websites als applicatiecomponent en het gebruik van services

**Vraagstelling:** In toenemende mate wordt de functionaliteit van backoffice / legacy applicaties ontsloten via webapplicaties. Daarbij wordt de functionaliteit van de BackOffice applicatie ook in vaak meerdere websites gebruikt en kan een website gebruik maken van externe services. Hoe modelleer je dit?

**Oplossing:** Aan de hand van onderstaand voorbeeld wordt één en ander beschreven:



Figuur 3-11: Websites en applicatieservices

De backoffice applicatie realiseert een service 'premieberekening'. Deze service wordt gebruikt door de twee websites. In deze situatie is backoffice functionaliteit ontsloten voor de webapplicaties.

In website 2 wordt gebruik gemaakt van de volgende externe services: 'Authenticatie' (bijv. Digid) en 'Routeberekening' (bijv. via GoogleMaps) . Deze laatste twee services worden extern betrokken.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Hier ligt een relatie met de good practice 'Koppeling tussen applicaties'.

## 4 Modelleren in de infrastructuur- / technologielaag

### 4.1 Overzicht good practices

In dit hoofdstuk zijn de onderstaande good practices opgenomen:

1. Infrastructuurservices.
2. Infrastructuurlandschap.
3. Onderscheid systeemsoftware en artefact.

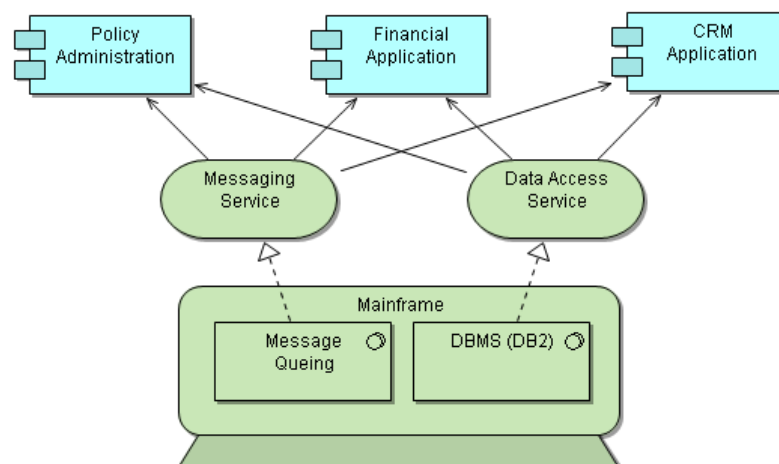
### 4.2 Infrastructuurservices

**Vraagstelling:** Welke services zou je op de infrastructuurlaag willen modelleren?

Vanuit het oogpunt van beheer is op de infrastructuurlaag vaak veel detailinformatie nodig: welke systemen zijn er en hoe zijn die precies met elkaar verbonden? De stap naar het beschrijven van voor de omgeving betekenisvolle services is niet triviaal: hoe begin je met het modelleren van services op de infrastructuurlaag en welke typen services moet je hierbij onderscheiden?

**Oplossing:** Beschrijf op de infrastructuurlaag de services waarmee de infrastructuur applicaties ondersteunt. Het is aan te raden hierbij onderscheid te maken tussen verwerkingsservices, opslagservices en communicatieservices. Essentieel bij het modelleren van de infrastructuurlaag is het abstraheren van interne details, bijvoorbeeld over implementatiedetails. Hiervoor kunnen domeinspecifieke talen gebruikt worden.

Maak hierbij gebruik van het Infrastructuurgebruik viewpoint uit het ArchiMateboek (Lankhorst et al., 2005, p. 187). Dit viewpoint geeft je de mogelijkheid te laten zien hoe applicaties ondersteund worden door software en hardware infrastructuur.



Figuur 4-1: Voorbeeld van infrastructuurservices

Hanteer de definitie zoals beschreven in ArchiMate (Lankhorst et al., 2005): een infrastructuurservice is een zichtbare eenheid van functionaliteit, geleverd door één of

meerdere nodes, aangeboden via goed gedefinieerde interfaces en betekenisvol voor de omgeving.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Niet van toepassing.

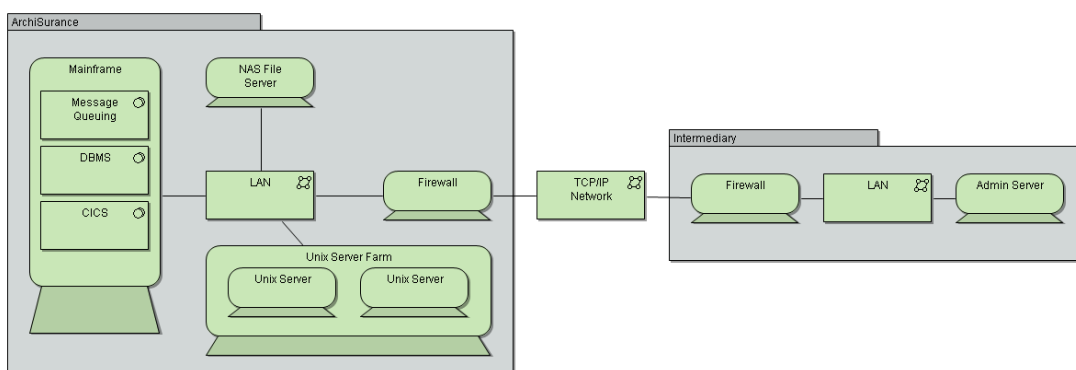
#### 4.3 Infrastructuurlandschap

**Vraagstelling:** Hoe modelleren je een infrastructuurlandschap in ArchiMate? Enterprise architectuur gebruik je om de essentie van architectuurdomeinen vast te leggen en de relaties tussen die domeinen expliciet te maken; echter op het niveau van infrastructuur zijn vaak implementatie details belangrijk. Welke aspecten moet je wel en niet modelleren in een infrastructuurlandschap in ArchiMate?

**Oplossing:** Zoals gebruikelijk is het essentieel om bij het modelleren van een infrastructuurlandschap uit te gaan van het doel en de doelgroep. In de meeste enterprise architectuur trajecten is het doel van een infrastructuurlandschap het in beeld brengen van de belangrijkste elementen (hardware en software) van de infrastructuur, hoe deze via netwerken verbonden zijn en wat hun geografische locatie is (zoals het hoofdkantoor en regiokantoren).

Laat in een ArchiMate infrastructuurlandschap alleen de belangrijkste fysieke systemen en netwerken zien en specificeer hierbij de essentiële ondersteunende software zoals operating systems, database management systemen en middleware.

Maak hierbij gebruik van het Infrastructuur viewpoint uit het ArchiMateboek (Lankhorst et al., 2005, p. 186). Dit viewpoint geeft je de mogelijkheid te laten zien welke essentiële hardware en software het infrastructuur landschap bepalen en hoe deze via netwerken verbonden zijn. Het is aan te raden in dit viewpoint infrastructuur elementen op basis van geografische locatie te groeperen en deze groepen expliciet te maken.



Figuur 4-2: Voorbeeld van een infrastructuurlandschap

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Niet van toepassing.

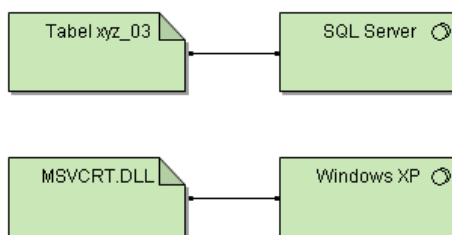
#### 4.4 Onderscheid systeemsoftware en artefact

**Vraagstelling:** Wat is het onderscheid tussen de ArchiMate-concepten 'Systeemsoftware' en 'Artefact'? Deze concepten worden in de praktijk op verschillende manieren gehanteerd.

**Oplossing:** Systeemsoftware en artefact zijn beide concepten uit de infrastructuurlaag van ArchiMate. Systeemsoftware maakt onderdeel uit van de gedragsconcepten, terwijl artefact een informatieconcept is. De definities (in het engels) van beide concepten zijn als volgt:

- *Systemsoftware*: System software represents a software environment for specific types of components and objects that are deployed on it in the form of artifacts
- *Artefact*: A physical piece of information that is used or produced in a software development process, or by deployment and operation of a system

Systeemsoftware is een specialisatie van een node, en wordt gebruikt om de software-omgeving te modelleren waarop artefacten gedeployed worden. Een artefact representeert een concreet fysiek element, en wordt gebruikt om (software) producten te modelleren, zoals source code, executables, scripts, database tabellen, berichten, documenten, specificaties, etc.



Figuur 4-3: Relatie tussen systeemsoftware en artefact

Modellering van een element (bijvoorbeeld software) als artefact legt de nadruk op het informatie-aspect, de fysieke kant (een bestand, een aantal regels code of een databasetabel). Modellering van een element als systeemsoftware legt de nadruk op het gedrag van dat element. Verder is er een “natuurlijke” relatie tussen artefacten en systeemsoftware, namelijk dat een artefact gedeployed wordt op systeemsoftware. Dit wordt gemodelleerd met een toekenningrelatie. Voorbeelden zijn een databasetabel die gedeployed wordt op een SQL Server database, of een dll-bestand dat op Windows XP geëxecuteerd wordt.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Niet van toepassing.

## 5 Modelleren over de grenzen van lagen heen

### 5.1 Overzicht good practices

In dit hoofdstuk zijn de onderstaande good practices opgenomen:

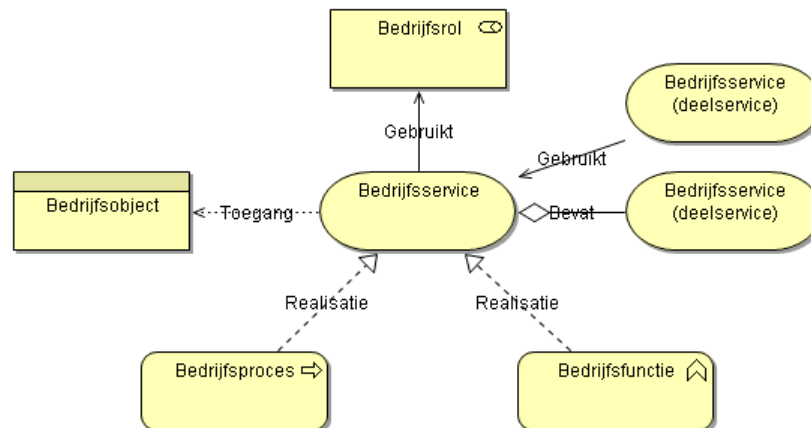
1. Veelgebruikte relaties.
2. Product en applicatieservices.
3. Ondersteuning van batch en online processen.
4. Servicebroker.
5. Domein.
6. Interne en externe services.
7. Noodzaak van het gebruik van services.
8. Vereenvoudigingen met behoud van consistentie.
9. Groepering.
10. Gegevensuitwisseling met externe organisaties in applicatielaag.
11. Weergeven van berichten in wachtrijen.
12. 'Nesten' en impliciet afgeleide relaties.
13. File transfer.

### 5.2 Veelgebruikte relaties

**Vraagstelling:** ArchiMate onderkent een behoorlijk aantal concepten en relaties. De vraag welke relatie gemodelleerd mag worden tussen een tweetal concepten is weliswaar precies gedefinieerd in ArchiMate, maar in de praktijk blijkt het lastig om de juiste keuze te maken binnen de toegestane mogelijkheden. De vraag is dan ook welke relaties tussen concepten in de meest voorkomende gevallen gemodelleerd zouden moeten worden. Hieronder geven we voor de meest voorkomende situaties aan welke relaties gemodelleerd zouden moeten worden. Om misverstanden te voorkomen: in onderstaande opsomming worden niet alle ArchiMate-relaties tussen concepten besproken, alleen de relaties die naar onze inschatting het vaakst gebruikt worden. Een compleet overzicht van alle relaties die mogelijk zijn tussen verschillende concepten is te vinden in het referentiemanual van ArchiMate.

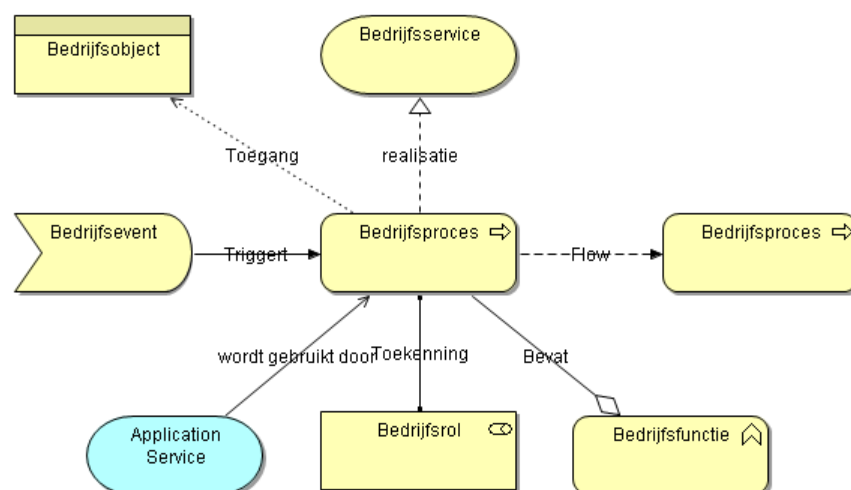
**Oplossing:** Hieronder volgt voor een aantal concepten (met name uit de bedrijfslaag en de applicatielaag) een overzicht van de meest gebruikte relaties vanuit of naar dat concept met andere concepten. Niet alle concepten worden hieronder opgesomd, alleen de meest gebruikte.

- **Bedrijfsservice:**
  - Een bedrijfsservice wordt gerealiseerd door een bedrijfsproces of een bedrijfsfunctie;
  - Een bedrijfsservice wordt gebruikt door een bedrijfsrol;
  - Een bedrijfsservice heeft toegang tot een bedrijfsobject (voorbeeld: een bedrijfsservice creëert, leest, wijzigt of vernietigt een bedrijfsobject);
  - Een bedrijfsservice kan bestaan uit andere bedrijfsservices;
  - Een bedrijfsservice kan andere bedrijfsservices gebruiken.



Figuur 5-1: Belangrijkste relaties tussen concepten

- **Bedrijfsproces:**
  - Een bedrijfsproces realiseert een bedrijfsservice;
  - Een bedrijfsproces wisselt gegevens uit met andere bedrijfsprocessen (via de flowrelatie);
  - Een bedrijfsproces wordt getriggerd door of triggert een bedrijfsevent, een bedrijfsfunctie of andere bedrijfsprocessen;
  - Een bedrijfsproces wordt toegekend aan een bedrijfsrol;
  - Een bedrijfsproces maakt onderdeel uit van een bedrijfsfunctie;
  - Een bedrijfsproces heeft toegang tot een bedrijfsobject (een bedrijfsproces creëert, leest, wijzigt of vernietigt een bedrijfsobject);
  - Een bedrijfsproces gebruikt applicatieservices.



Figuur 5-2: Belangrijkste relaties tussen concepten

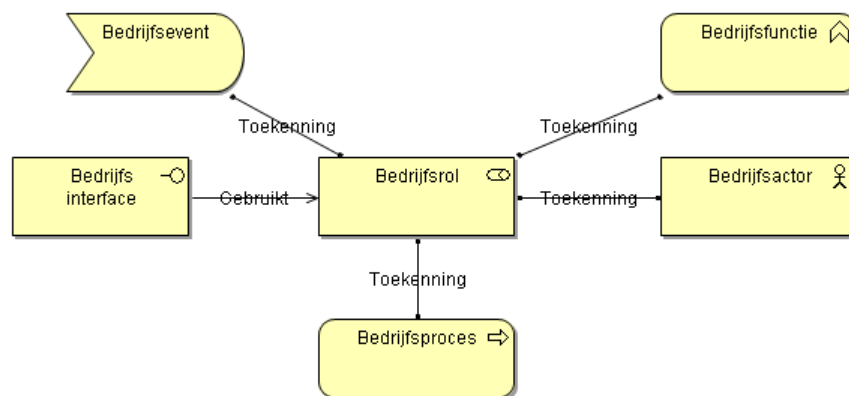
- **Bedrijfsactor:**
  - Een bedrijfsrol kan toegekend worden aan een bedrijfsactor.





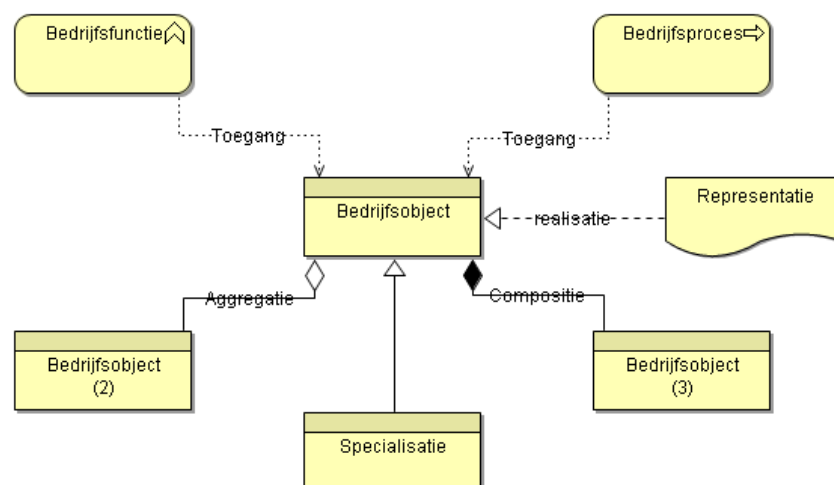
Figuur 5-3: Relatie tussen bedrijfsrol en bedrijfsactor

- Bedrijfsrol:
  - Een bedrijfsrol kan toegekend worden aan een bedrijfsactor;
  - Een bedrijfsrol kan toegekend worden aan een bedrijfsproces, bedrijfsfunctie of bedrijfsevent.
  - Een bedrijfsrol kan gebruik maken van een bedrijfsinterface.



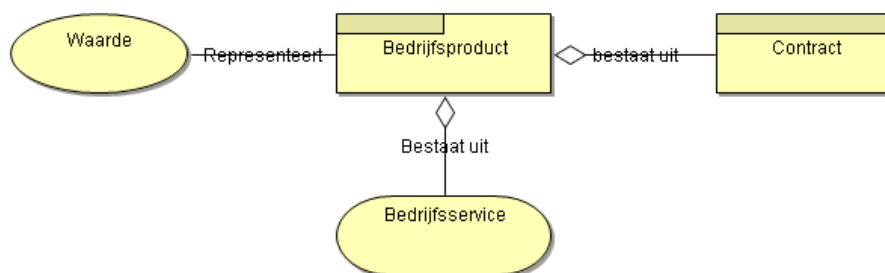
Figuur 5-4: Belangrijkste relaties tussen concepten

- Bedrijfsobject:
  - Een bedrijfsobject wordt gecreëerd, gelezen, gewijzigd of verwijderd door een bedrijfsproces of bedrijfsfunctie (via toegangsrelatie);
  - Een bedrijfsobject kan specialisaties hebben;
  - Een bedrijfsrepresentatie realiseert een bedrijfsobject;
  - Een bedrijfsobject kan verwijzen naar andere objecten (aggregatierelatie);
  - Een bedrijfsobject kan andere objecten bevatten (compositierelatie).



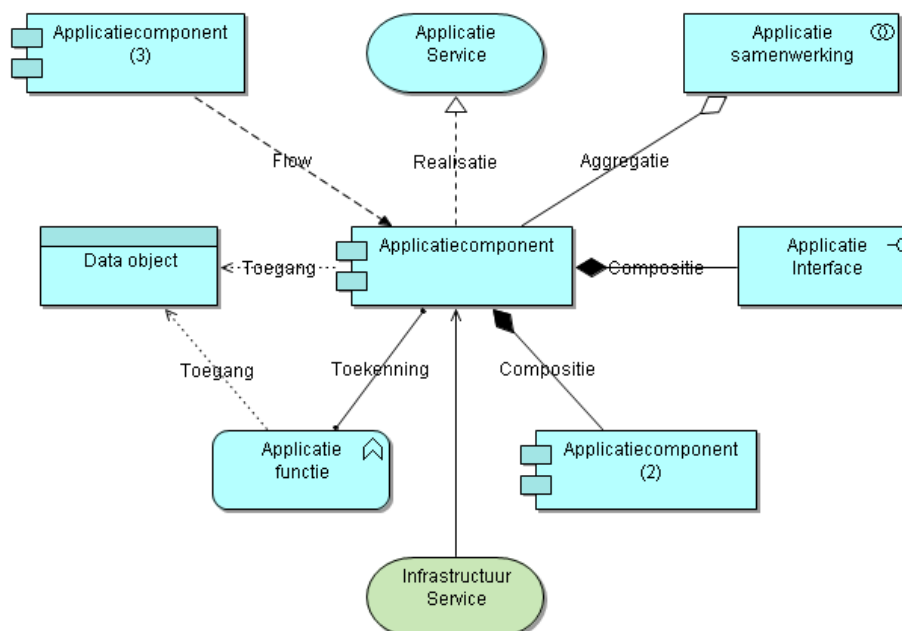
Figuur 5-5: Belangrijkste relaties tussen concepten

- Bedrijfsproduct:
  - Een bedrijfsproduct bestaat uit services en contract(en);
  - Een bedrijfsproduct representeert een waarde.



Figuur 5-6: Belangrijkste relaties tussen concepten

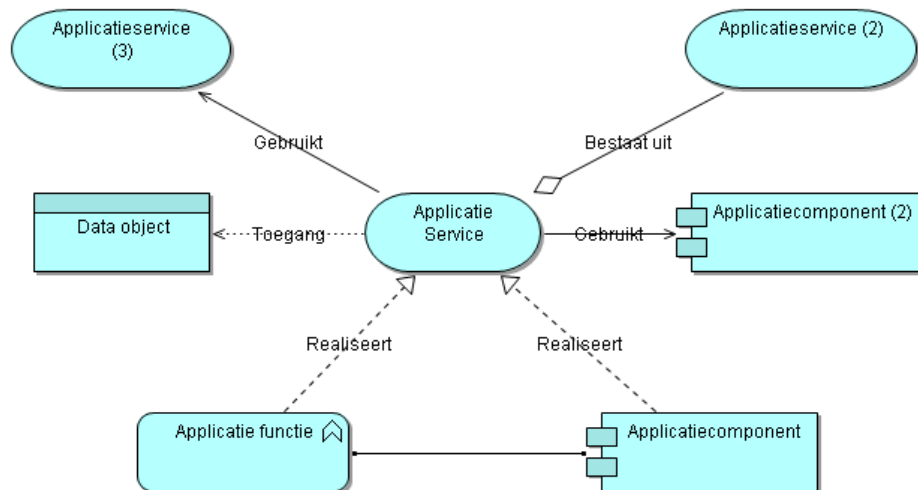
- Applicatiecomponent:
  - Een applicatiecomponent realiseert een applicatieservice;
  - Een applicatiecomponent heeft een of meerdere applicatie-interfaces;
  - Een data-object wordt gecreëerd, gelezen, gewijzigd of verwijderd door een applicatiecomponent of applicatiefunctie;
  - Een applicatiecomponent kan onderdeel zijn van een applicatiesamenwerking (via aggregatierelatie);
  - Een applicatiecomponent kan bestaan uit meerdere applicatiecomponenten (via compositierelatie);
  - Een applicatiecomponent kan toegekend worden aan een applicatiefunctie;
  - Een applicatiecomponent gebruikt infrastructuurservices;
  - Tussen applicatiecomponenten kunnen gegevenstromen bestaan (flowrelatie).



Figuur 5-7: Belangrijkste relaties tussen concepten

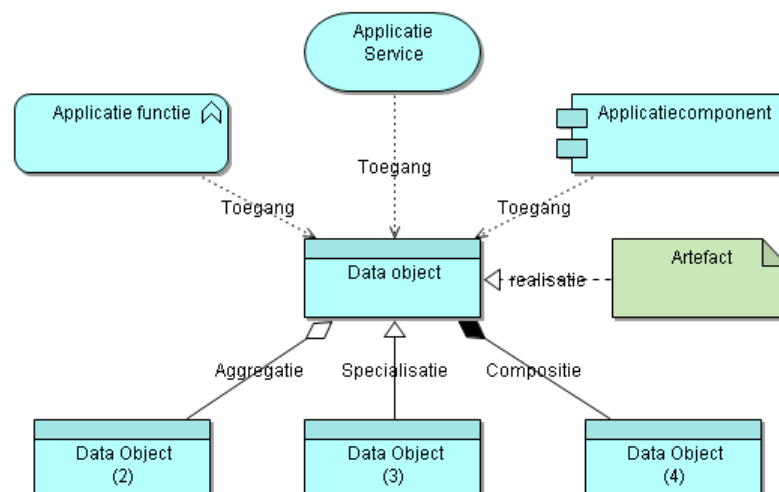
- Applicatieservice:
  - Een applicatieservice wordt gerealiseerd door een applicatiecomponent of een applicatiefunctie;

- Een applicatieservice wordt gebruikt door een applicatiecomponent;
- Een applicatieservice heeft toegang tot een data-object (een applicatieservice creëert, leest, wijzigt of vernietigt een data-object)
- Een applicatieservice kan bestaan uit andere applicatieservices., en kan andere applicatieservice gebruiken.



Figuur 5-8: Belangrijkste relaties tussen concepten

- Data-object:
  - Een data-object wordt gecreëerd, gelezen, gewijzigd of verwijderd door een applicatiecomponent, applicatieservice of applicatiefunctie (via toegangsrelatie);
  - Een data-object kan specialisaties hebben;
  - Een artifact realiseert een data-object;
  - Een data-object kan verwijzen naar andere data-objecten (aggregatierelatie);
  - Een data-object kan andere data-objecten bevatten (compositierelatie).



Figuur 5-9: Belangrijkste relaties tussen concepten

**Gevolgen:** Let op, dit zijn een aantal veelvoorkomende situaties. Het gebruik van relaties binnen ArchiMate is breder dan hierboven aangegeven. Een volledig overzicht is te vinden in de Architecture Language Reference Manual.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Niet van toepassing.

### 5.3 Product en applicatieservices

**Vraagstelling:** Is het niet vreemd dat in een product naast bedrijfsservices ook applicatie- en infrastructuurservices zitten? Een klant kan per definitie toch geen applicatieservices afnemen? Dus als een applicatieservice bijna direct wordt gebruikt door een klant, dan zou je daar toch een bedrijfsservice tussen moeten plaatsen? In sommige voorbeelden, bijvoorbeeld in de Architecture Language Reference Manual, wordt een product namelijk samengesteld uit applicatie- en bedrijfsservices.

**Oplossing:** De Architecture Language Reference Manual bevat inderdaad voorbeelden waar klanten rechtstreeks applicatieservices gebruiken. Het is beter om daar een bedrijfsservice tussen te zetten. In de productbeschrijving van de Architecture Language Reference Manual (p. 26) worden de applicatieservices 'Account status' en 'Money transfer' dan bedrijfsservices.

De koppeling tussen een bedrijfsservice en een applicatieservice verloopt dan via een bedrijfsproces die de bedrijfsservice ondersteunt, en op zijn beurt ondersteund wordt door een applicatieservice. Daarmee geldt de afgeleide relatie dat de bedrijfsservice een applicatieservice gebruikt.

**Gevolgen:** Overwogen wordt of de aggregatierelatie tussen product en applicatieservice verwijderd wordt.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Niet van toepassing.

### 5.4 Ondersteuning van batch en online processen

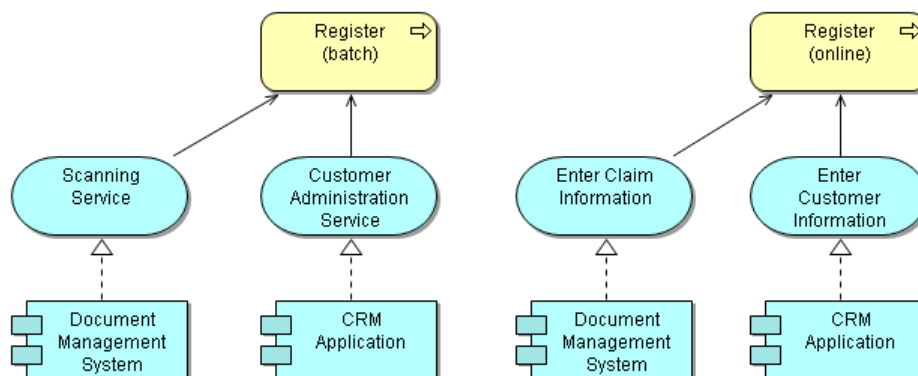
**Vraagstelling:** Hoe modelleer je in ArchiMate twee varianten van de ondersteuning van hetzelfde proces, waarbij één via een batch verloopt en de ander handmatig (online) verloopt?

**Oplossing:** Onderscheid twee varianten van het bedrijfsproces: één voor de handmatige (online) verwerking en één voor de batch verwerking. Hoewel de resultaten van de processen hetzelfde kunnen zijn, biedt de handmatige (online) verwerking meer ruimte voor uitzonderingen en kunnen vaak fijnmaziger processtappen onderscheiden worden. In beide procesbeschrijvingen is het verstandig bij het vaststellen van de 'grootte' van acties aan te houden dat acties *eenheden zijn van tijd, plaats en handeling*. Dit zal in de twee varianten van het bedrijfsproces waarschijnlijk verschillend zijn.

Beschrijf op basis van de twee varianten van het bedrijfsproces de applicatieservices die deze processen ondersteunen. Ga hierbij uit van de acties die gemodelleerd zijn in het bedrijfsproces: de applicatieservices voor beide varianten kunnen dus verschillend zijn!

Vaak zal het zo zijn dat de applicatieservices voor het ondersteunen van een batch proces meer grofmazig zijn dan applicatieservices voor het ondersteunen van een online proces, omdat in het laatste geval meer fijnmazige acties onderscheiden worden. Let wel, het is goed mogelijk dat de *applicatiefuncties* die de verschillende versies van de applicatieservices realiseren wél gemeenschappelijk zijn: dit wijst op hergebruik van functionaliteit in de batch en online verwerking.

Maak hierbij gebruik van het Applicatiegebruik viewpoint uit het ArchiMateboek (Lankhorst et al., 2005, p. 184). Dit viewpoint geeft je de mogelijkheid te laten zien hoe bedrijfsprocessen ondersteund worden door applicaties, via applicatieservices.



Figuur 5-10: Voorbeeld van een batch en online (handmatig) variant van hetzelfde proces

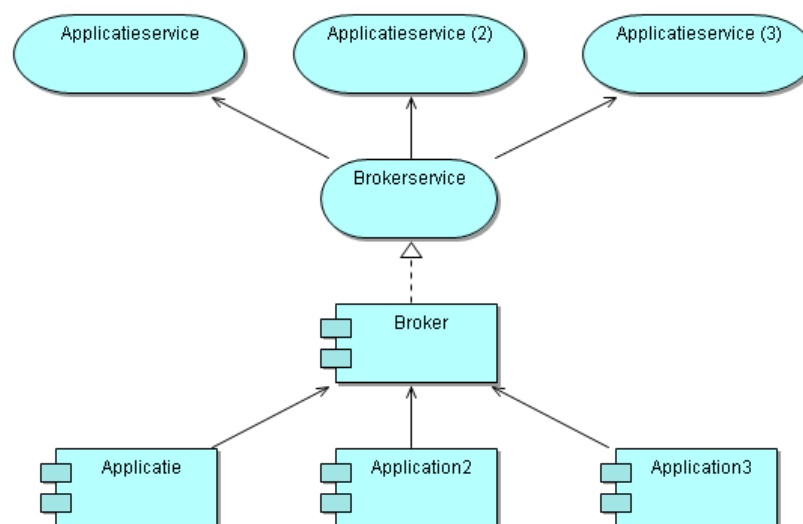
**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Zie ook de good practices 'Bedrijfservice, bedrijfsfunctie en bedrijfsproces', 'Afbakening bedrijfsproces' en 'Viewpoints voor procesmodellering'.

## 5.5 Servicebroker

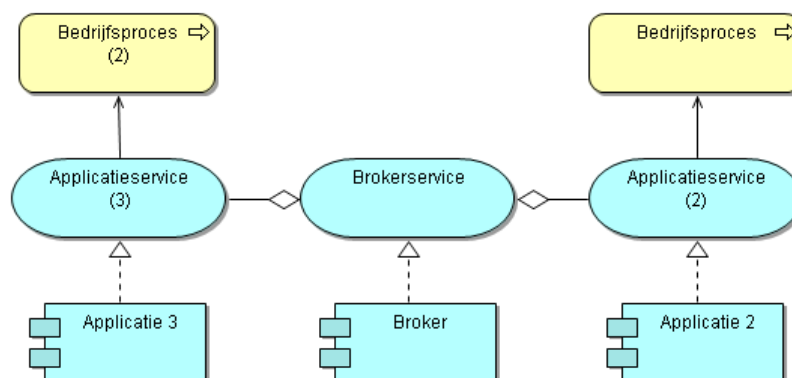
**Vraagstelling:** Hoe modelleer je een servicebroker (of een soortelijke component, bijvoorbeeld een workflowhandler of een middleware-achtige component) zodat enerzijds duidelijk is dat via de broker alle services geleverd worden, maar dat daarbij zichtbaar blijft welke applicatie welke service (via de broker) levert. Het doel van een broker is het vereenvoudigen van de vele relaties tussen verschillende applicaties. Soms wordt de volgende modellering gebruikt:



Figuur 5-11: Servicebroker in relatie tot applicaties - niet gewenst

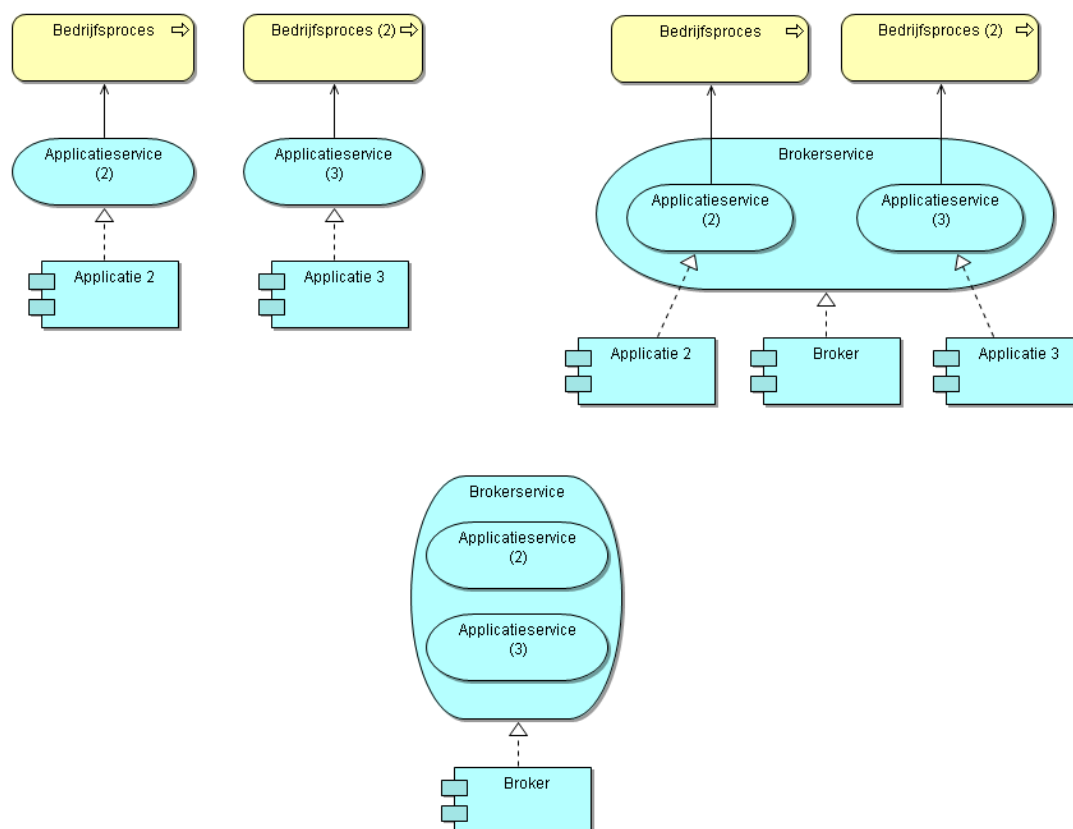
Nadeel van deze manier van modelleren is dat niet meer zichtbaar is welke applicatie welke service levert. Ook is het wenselijk om naar keuze de tussenliggende broker (en zijn service) te laten zien dan wel weg te laten. Dit is met bovenstaande modellering niet mogelijk.

**Oplossing:** De kern van de oplossing is dat enerzijds de relatie tussen applicaties en applicatieservices gelegd wordt (de applicatie realiseert de bijbehorende applicatieservice), en dat anderzijds een relatie tussen de applicatieservices en de brokerservice wordt gelegd (de brokerservice “bevat” de applicatieservices). De bovenstaande situatie wordt dan als volgt gemodelleerd:



Figuur 5-12: Servicebroker in relatie tot applicaties - gewenst

Door nu verschillende views hiervan te generen (en daarin bepaalde relaties weg te laten dan wel als nesting te tonen), kunnen verschillende accenten gelegd worden. In de volgende figuren is achtereenvolgens zichtbaar de relatie tussen applicaties en services zonder broker, de relatie tussen applicaties en services met broker, en de rol van de broker afzonderlijk.



Figuur 5-13: Servicebroker in relatie tot applicaties - views

**Gevolgen:** Door een broker en de brokerservice op bovenstaande manier te modelleren wordt het mogelijk om naar keuze de broker wel of niet te laten zien, hetgeen de inzichtelijkheid van het model voor de verschillende doelgroepen ten gunste komt.

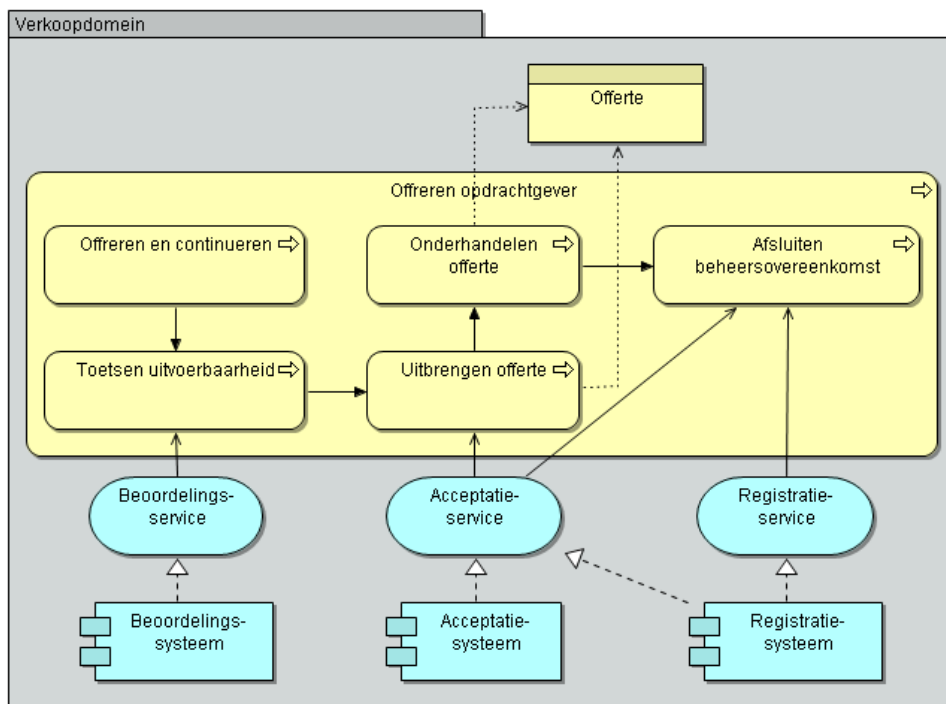
**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** De servicebroker is een onderdeel van het applicatielandschap. Zie hiervoor de good practice 'Applicatielandschap'.

## 5.6 Domein

**Vraagstelling:** Hoe modelleer ik een 'domein' in ArchiMate? Een 'domein' definiëren we als een bepaalde afbakening ten behoeve van de indeling van concepten. Dit zouden bedrijfs-, applicatie-, informatie- of technische domeinen kunnen zijn.

**Oplossing:** In ArchiMate is de grouping relatie geschikt om domeinen te groeperen. De relatie kan vervolgens een naam worden gegeven. In onderstaand voorbeeld is (een deel van) het verkoopdomein weergegeven.



Figuur 5-14: Voorbeeld van een verkoopdomein

**Gevolgen:** ArchiMate maakt onderscheid tussen de concepten en de relaties. Grouping is een relatie. De grouping relatie wil je kunnen hergebruiken in andere views (zo mogelijk zonder de elementen expliciet op te nemen, dus alleen door de 'grouping' op te nemen). ArchiMate laat in het midden of bij het gebruik van grouping daadwerkelijk relaties worden gelegd. De bestaande tools ondersteunen dat niet. Het gebruik van de grouping relatie betekent dat er flexibel met domeinen omgegaan kan worden. Elk denkbaar domein kan op deze manier vormgegeven worden. Om wildgroei te voorkomen is het raadzaam om vooraf te definiëren welke domeinen onderscheiden worden binnen de organisatie en welke concepten daar deel van uitmaken (aggregatierelatie).

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Zie ook de good practice 'applicatielandschap'.

## 5.7 Interne en externe services

**Vraagstelling:** ArchiMate maakt onderscheid tussen services die gebruikt worden binnen de eigen laag en door de bovenliggende laag.

De 'eigen laag' betreft de businesslaag voor bedrijfsservices, de applicatielaag voor applicatieservices en de infrastructuurlaag voor de infrastructuurservice. ArchiMate noemt dit interne services.

De 'bovenliggende laag' is voor de infrastructuurservices de applicatielaag, voor de applicatieservices de businesslaag en voor de bedrijfsservices de omgeving. ArchiMate noemt dit externe services. De begrippen intern en extern zijn dus relatieve begrippen ten opzichte van de laag waar de service in gerealiseerd wordt.

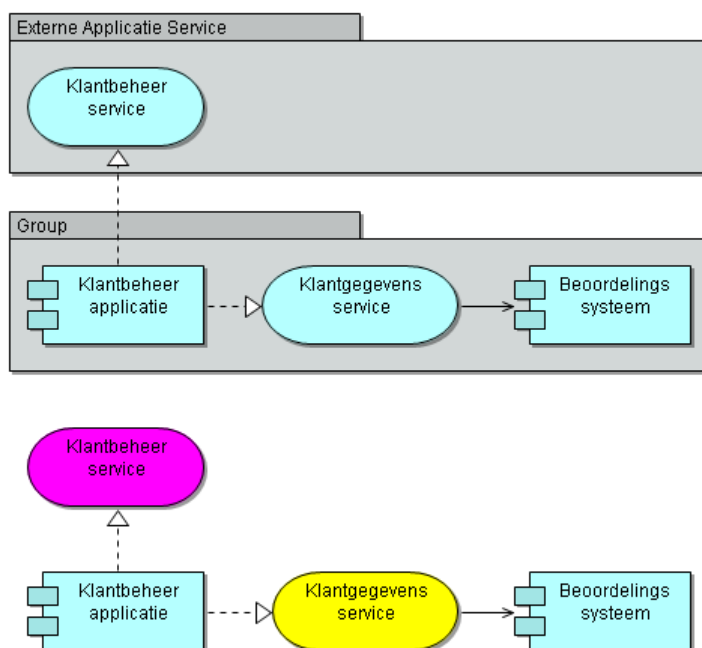
Hoe breng je dit verschil in beeld? Zie ook Enterprise Architecture at work, p. 86.



**Oplossing:** Het definiëren van services die gebruikt worden binnen de eigen laag ('interne services') is wat vaak onder Service Oriëntatie verstaan wordt. De gemodelleerde services komen dan meestal overeen met daadwerkelijke geïmplementeerde / te implementeren services (met name als het gaat om applicatieservices en infrastructuurservices). Bij services die door de bovenliggende laag gebruikt worden ('externe services') gaat het om betekenisvolle functionaliteit die de bovenliggende laag ondersteunt; dit betreft niet altijd een daadwerkelijk geïmplementeerde service (in de zin van een daadwerkelijk extern aanroepbare functionaliteit).

Als good practice bevelen we aan om, wanneer het onderscheid tussen interne en externe services belangrijk is, interne en externe services als twee aparte groepen te behandelen, en in views ook visueel te onderscheiden. Vaak zullen aan externe services andere eisen gesteld worden dan aan interne services, en zullen er andere eigenschappen van vastgelegd moeten worden. Ook zal de service zelf vaak anders ontworpen zijn, al naar gelang deze bedoeld is voor ondersteuning binnen de eigen laag of de bovenliggende laag.

Figuur 5-15 geeft een voorbeeld hoe dat onderscheid visueel kan worden weergegeven: de interne en externe services worden onderscheiden door ze in aparte lagen (groepering) te plaatsen<sup>1</sup>. In dit voorbeeld is de Klantbeheerservice een externe service en staat dus in de desbetreffende (tussen)laag weergegeven. De Klantgegevensservice wordt gebruikt op de applicatielaag zelf en is dus een interne service. Een andere mogelijkheid is om interne en externe services een andere kleur en/of vorm te geven.



Figuur 5-15: In- en externe services

**Gevolgen:** Maak expliciet onderscheid tussen interne en externe services.

<sup>1</sup> Merk op dat in Figuur 5-15 een vereenvoudiging is toegepast door de applicatiefunctie weg te laten - zie de good practice "Vereenvoudigen met behoud van consistentie".

**Alternatieven:** In deze good practice wordt onderscheid gemaakt tussen interne en externe services. Er kunnen ook redenen zijn om niet dit onderscheid, maar een ander onderscheid te maken, bijvoorbeeld tussen bedrijfskritische en niet-bedrijfskritische services, of tussen services met een gouden, zilveren of bronzen service level agreement. Voor een dergelijk onderscheid geldt een soortgelijke benadering als hierboven, namelijk dat via groepering, kleur of vorm dit onderscheid benadrukt kan worden. Het modelleren- en visualiseren doel bepaalt feitelijk welk onderscheid zo relevant is dat het afzonderlijk belicht moet worden.

**Relaties met andere good practices:** Hier ligt een relatie met de good practice ‘Koppeling tussen applicaties’.

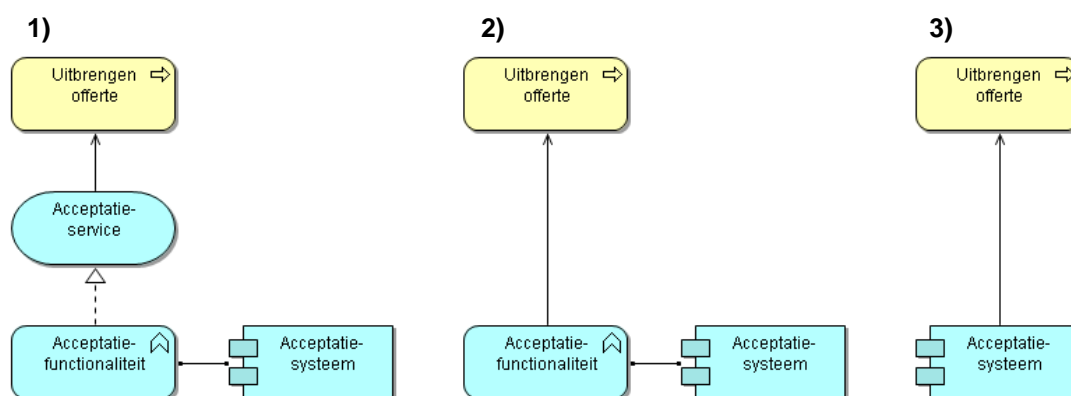
## 5.8 Noodzaak voor het gebruik van services

**Vraagstelling:** Moet ik persé services gebruiken binnen ArchiMate? In mijn huidige situatie hebben wij geen services geïmplementeerd. Ook kijken wij niet vanuit de servicegedachte tegen de wereld aan.

**Oplossing:** Het is niet verplicht om binnen ArchiMate services te gebruiken. Als die niet onderscheiden worden en ook in de toekomst niet voorzien zijn, kunnen de services eventueel weggelaten worden.

**Gevolgen:** Als er geen services worden onderscheiden, worden er ook geen services gemodelleerd. De regels ten aanzien van de good practice ‘Vereenvoudigingen met behoud van consistentie’ moeten dan gebruikt worden om de concepten van ArchiMate consistent te blijven gebruiken.

In plaats van dat een proces een service gebruikt (1), gebruikt deze dan een applicatiefunctie (2) of, bij nog verdere vereenvoudiging, een applicatie (3).



Figuur 5-16: Gebruik van applicatiefunctie in plaats van applicatieservice

**Alternatieven:** Ook als er momenteel geen services onderscheiden worden, kan de servicegedachte wel helpen om een andere kijk op de wereld te hebben. Dit kan tot nieuwe inzichten leiden. Door de wereld wel vanuit de servicegedachte te bekijken komt men vaak tot andere indelingen en verantwoordelijkheden.

Indien er voor de toekomst wel voorzien wordt om de servicegedachte te adopteren, kan die toekomstige situatie met behulp van services gemodelleerd worden.

**Relaties met andere good practices:** Zie ook de good practice ‘Vereenvoudigingen met behoud van consistentie’.

## 5.9 Vereenvoudigingen met behoud van consistentie

**Vraagstelling:** Hoe kan ik mijn views vereenvoudigen, terwijl ik weet dat de consistentie bewaard is?

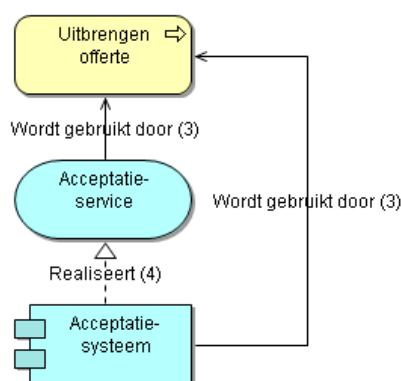
**Oplossing:** Binnen ArchiMate is het mogelijk om indirecte relaties af te leiden door composities van relaties. Dit wordt ook beschreven in paragraaf 5.7 van het boek EA at work. Dit blijkt een voor het weergeven van architectuur onmisbaar instrument om details te kunnen weglaten en toch consistentie tussen modellen te handhaven.

Voor de *structurele* relaties is een krachtige compositieregel ontwikkeld die gebaseerd is op de *sterkte* van de relatie. Het blijkt dat in een keten van concepten de ‘zwakste’ relatie in de keten de algehele relatie bepaalt tussen de ‘eindconcepten’ van de keten. De onderstaande tabel toont de sterkte van de structurele relaties. Figuur 1 laat een voorbeeld zien van de toepassing van deze compositieregel.

In dit voorbeeld is de afgeleide relatie tussen het proces Registreren en de applicatiefunctie Beheren Klantgegevens een *wordt gebruikt door* relatie aangezien deze een lager gewicht heeft dan een *realisatie*.

Tabel Sterkte van structurele relaties

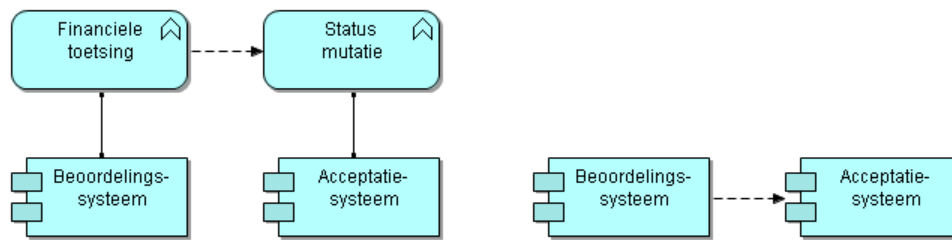
Relatie	Sterkte
association	1
access	2
used by	3
realisation	4
assignment	5
aggregation	6
composition	7



Figuur 5-17: Voorbeeld van een afgeleide relatie

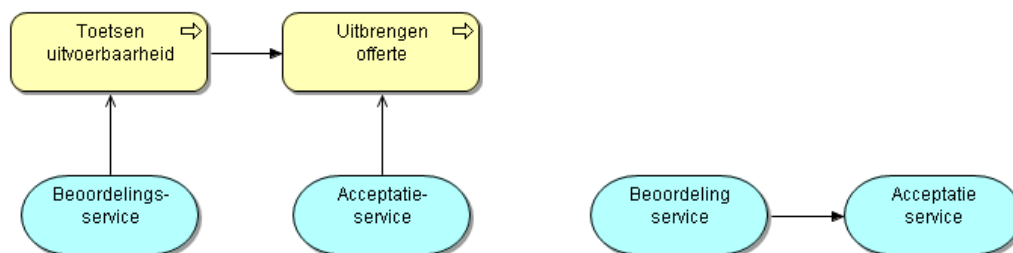
Voor de twee dynamische relaties zijn de volgende regels van toepassing:

- Een triggering- of flowrelatie tussen gedragselementen (b.v. processen of functies) mag worden overgezet naar structurele elementen (b.v. business actoren of applicatiecomponenten) aan welke ze zijn toegekend.



Figuur 5-18: Voorbeeld van een afgeleide relatie

- Een triggering- of flowrelatie tussen gedragselementen mag worden overgezet naar de services die ze realiseren.



Figuur 5-19: Voorbeeld van een afgeleide relatie

**Gevolgen:** Views kunnen op deze manier vereenvoudigd worden naar een detailniveau dat voor het doel van de view relevant is, zonder dat de consistentie verloren gaat. Niet elke tool ondersteunt deze vereenvoudiging. Het gevolg daarvan is dat er dan redundante relaties gaan ontstaan. In bovenstaand voorbeeld van de services betekent dat, dat de bestaande relatie tussen de twee processen in de tool worden vastgelegd (linkerdeel). Als in een andere view het verband wordt aangegeven tussen de services (rechterdeel), maakt de tool een extra triggeringrelatie aan tussen deze twee services.

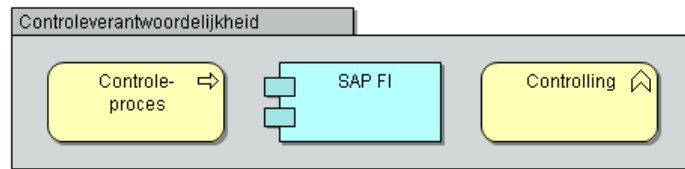
**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Deze vereenvoudigingsregels kunnen voor elke good practice van toepassing zijn. Zie bijvoorbeeld de good practice 'Koppeling tussen applicaties'.

## 5.10 Groepering

**Vraagstelling:** Welke voorzieningen kent ArchiMate om groepering weer te geven?

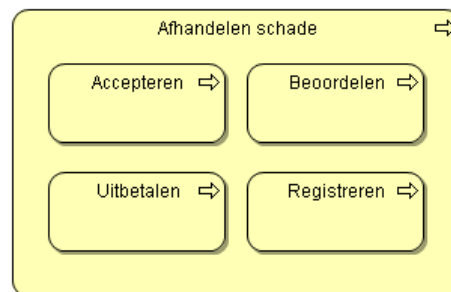
**Oplossing:** ArchiMate kent een apart concept voor groepering (feitelijk een groeperingsrelatie), zoals hieronder weergegeven. Met deze groepering kunnen willekeurige verzamelingen objecten (grafisch) gegroepeerd worden.



Figuur 5-20: Groeperingsconcept

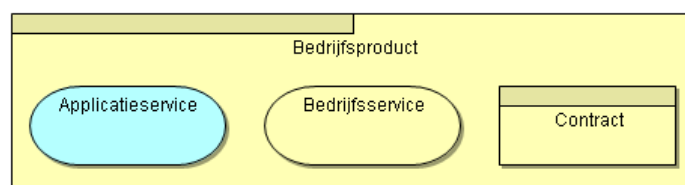
Deze grafische groepering heeft echter voor de meeste situaties onvoldoende semantiek.

ArchiMate kent daarnaast diverse andere groeperingsmechanismen. Zo kunnen vrijwel alle concepten hiërarchisch gegroepeerd worden met compositie- of aggregatierelaties. Daarmee kan bijvoorbeeld uitgedrukt worden dat een proces bestaat uit deelprocessen, of dat een organisatie bestaat uit afdelingen. Vaak worden hiermee gelijksoortige concepten gegroepeerd.



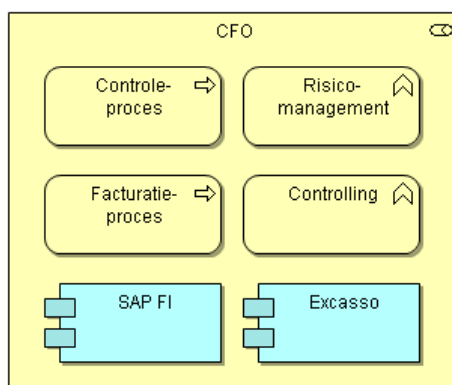
Figuur 5-21: Groepering van gelijksoortige concepten

Ook kunnen met deze relaties verschillende typen concepten gegroepeerd worden, bijvoorbeeld een bedrijfsproduct dat een combinatie vormt van bedrijfs- en applicatieservices en een contract (met aggregatie als onderliggende relatie).



Figuur 5-22: Groepering van verschillende typen concepten

Ook is het in ArchiMate mogelijk om met behulp van de associatierelatie willekeurige concepten te groeperen. Een voorbeeld hiervan is het modelleren van eigenaarschap of verantwoordelijkheid, waarbij een bepaalde rol eigenaar of verantwoordelijk is van een aantal applicaties, processen en bedrijfsfuncties. Deze concepten worden via een associatierelatie verbonden met de rol (waarbij de relatie getypeerd kan worden als “eigenaarschap”).



Figuur 5-23: Groepering via associatierelatie

Op deze manier kunnen willekeurige groeperingen gemaakt en getoond worden.

Vaak wordt ook de term “domein” gehanteerd om zaken te groeperen. Voorbeelden zijn het domein “Klant”, het domein “Leven”, etc. In veel gevallen kan dan een concept als bedrijfsfunctie, rol of applicatiefunctie gebruikt worden als domeinelement, waar binnen de overige elementen gegroepeerd worden.

Niet elke groepering hoeft overigens expliciet gemodelleerd te worden. In veel gevallen is een groepering niets anders dan een view vanuit een bepaalde invalshoek. Denk bijvoorbeeld aan een overzicht van alle processen die ondersteund worden door applicaties die vallen onder de verantwoordelijkheid van één specifieke rol. Een dergelijke groepering is af te leiden uit de relaties die er vanuit de rol via de applicaties lopen naar de betreffende processen.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Eén van de mogelijke uitbreidingen van de ArchiMate-taal betreft een generiek groeperingsconcept; het uitbreiden van de taal met een dergelijk concept lost uiteraard het bovenstaande probleem op. Mogelijk wordt een dergelijk concept opgenomen in een van de komende versies van de ArchiMate-taal.

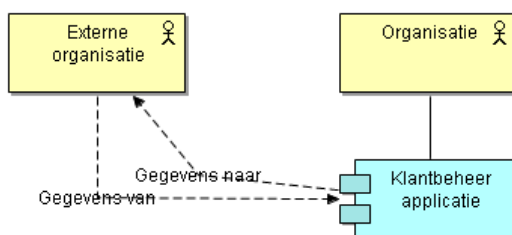
Bij gebruik van een ondersteunende tool is het in een aantal gevallen mogelijk om het metamodel van de tool uit te breiden met nieuwe concepten en relaties. In dat geval is het mogelijk om een nieuw groeperingsconcept toe te voegen, buiten de eigenlijke ArchiMate-taal om. Bij uitbreiding van de ArchiMate-taal met een groeperingsconcept dient dan uiteraard wel een conversie plaats te vinden.

**Relaties met andere good practices:** Hier ligt een relatie met de good practice ‘Domein’.

## 5.11 Gegevensuitwisseling met externe organisaties in applicatielaag

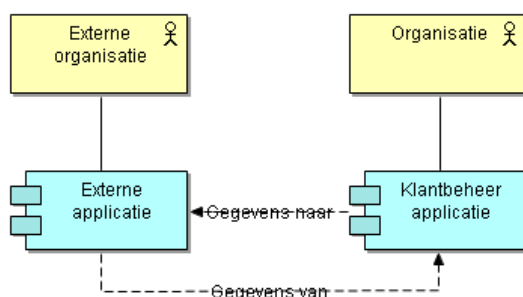
**Vraagstelling:** Hoe modelleer je gegevensuitwisseling met externe organisaties op de applicatielaag? Op applicatieniveau zijn er immers applicatiefuncties die deze gegevensuitwisseling realiseren. Maar moet je dan ook de externe applicatie in het applicatielandschap opnemen?

**Oplossing:** Een eerste alternatief is om een flowrelatie rechtstreeks van de eigen applicatie naar de externe organisatie te modelleren. Hierbij wordt dus de externe applicatie niet opgenomen.



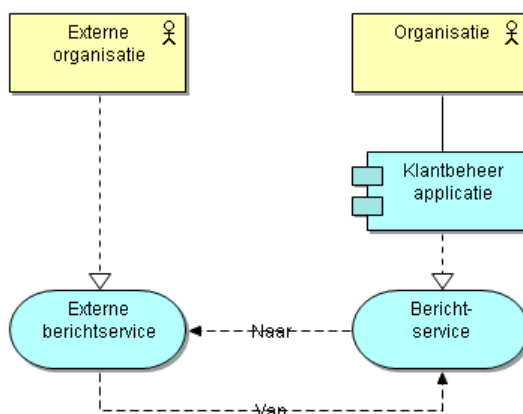
Figuur 5-24: Flowrelatie tussen eigen applicatie en externe organisatie

Hiermee wordt voorkomen dat externe applicaties in het applicatielandschap opgenomen worden. Bij gegevensuitwisselingen met diverse externe organisaties zou het aantal externe applicaties al gauw toenemen waardoor het applicatielandschap onoverzichtelijk wordt. Zie onderstaand voorbeeld:



Figuur 5-25: Flowrelatie tussen eigen applicatie en applicatie van externe organisatie

Een tweede alternatief is om een externe applicatieservice te modelleren. Hierbij wordt de gegevensuitwisseling tussen applicatieservices gemodelleerd.



Figuur 5-26: Flowrelatie tussen applicatieservice van eigen applicatie en die van externe organisatie

Indien gewenst kunnen de flowrelaties tussen de services vervangen worden door een data-object; geschreven door de ene service, gelezen door de andere service.

**Gevolgen:** Niet van toepassing.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Hier ligt een relatie met de good practice ‘Koppeling tussen applicaties’.

## 5.12 Weergeven van berichten in wachtrijen

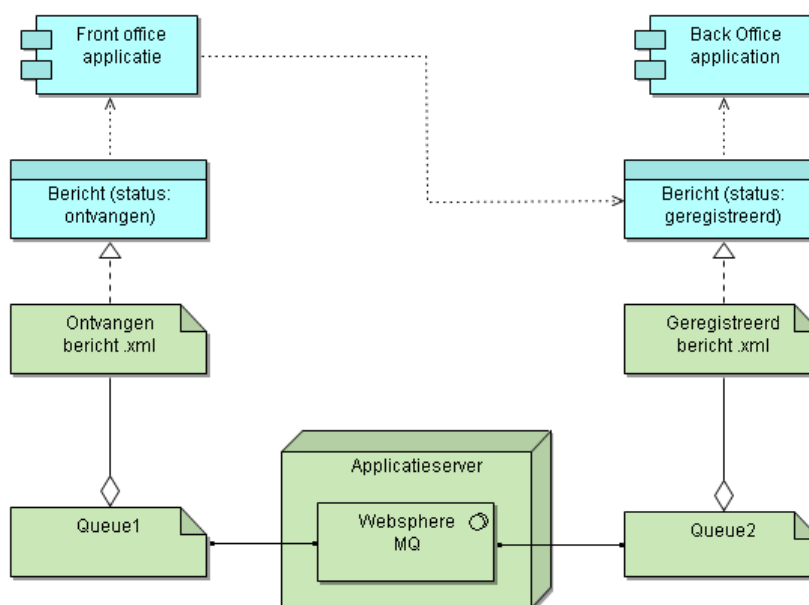
**Vraagstelling:** Ontvangen berichten worden in stappen verwerkt. Bij elke stap verandert de status van het bericht. Voor elke status bestaat een aparte wachtrij in MQ-series voor verdere bewerking door dezelfde of een andere applicatie. Inzicht is nodig in welke rij een bericht met een bepaalde status is vastgelegd ten behoeve van welke applicatie. Hoe modelleren je dat?

**Oplossing:** Onderstaande oplossing biedt een patroon voor de bovenstaande vraag. Het is geen expliciet voorbeeld.

De kern van de oplossing is erin gelegen dat in de infrastructuurlaag elke queue als artefact wordt gemodelleerd. Deze queues zijn dan met een toekenningsrelatie verbonden met de queue-manager. Deze queue-manager is systeemsoftware die op een node draait.

Het bericht met de betreffende status is ook een artefact, dat aan de betreffende queue is verbonden met een aggregatie-relatie. Deze artefacten realiseren dan het data-object in de applicatielaag.

Er kunnen zoveel queues, berichten en data-objecten worden gemodelleerd als nodig is. In het voorbeeld hieronder staan er twee, maar dat is met elk gewenst aantal uit te breiden. Voor de queues worden doorgaans technischere namen gehanteerd dan in het voorbeeld.



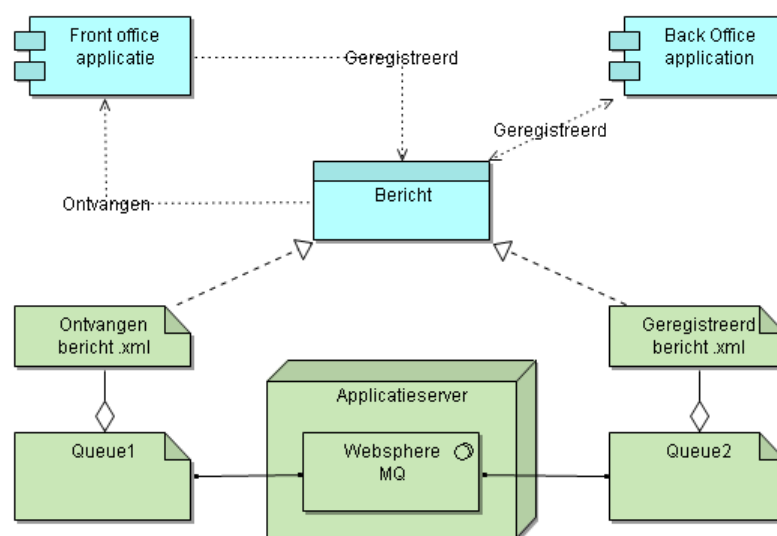
Figuur 5-27: Verschillende berichten in verschillende queues



**Gevolgen:** Wanneer vele queues en berichten op deze manier gemodelleerd moeten worden kan het model onoverzichtelijk worden. Ook wordt het bericht nu meerdere keren getoond in de applicatielaag, wat kan suggereren dat het om meerdere verschillende data-objecten gaat. Dat is hier echter niet het geval. Het is logisch gezien steeds hetzelfde data-object, maar met elke keer een andere status. Fysiek zijn het wel steeds andere XML-berichten.

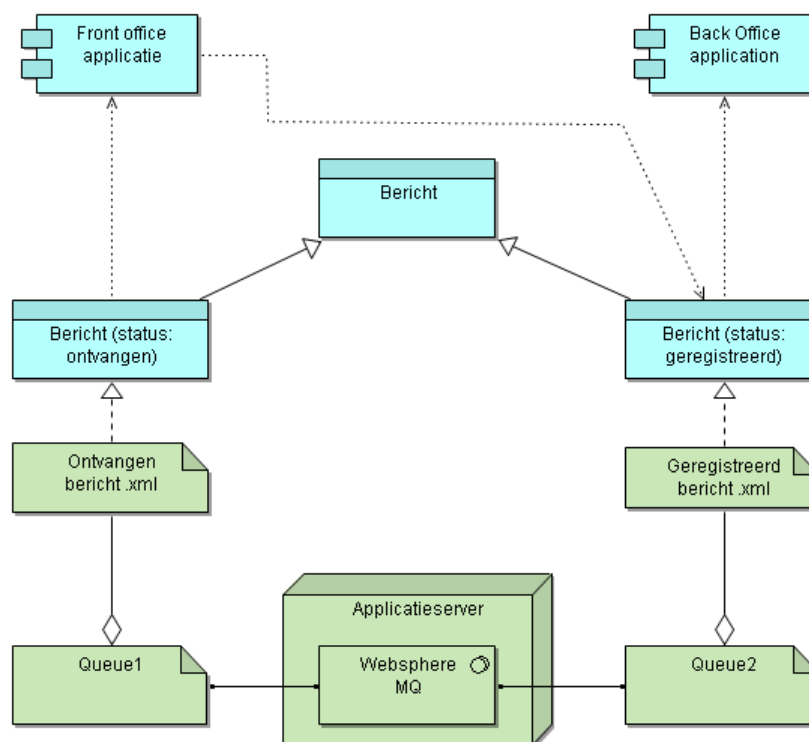
**Alternatieven:** Wanneer in het model duidelijk moet worden dat het inderdaad steeds om hetzelfde data-object gaat, zijn er twee alternatieven.

Het eerste alternatief laat het data-object één keer zien. De specifieke status van het bericht komt dan tot uitdrukking in de access-relatie tussen het data-object en de applicatie. Bij deze relatie wordt dan vermeld om welke status het gaat.



Figuur 5-28: De status in de accessrelatie

Een andere mogelijkheid is om de berichten met een verschillende status als een specialisatie van het oorspronkelijke bericht weer te geven. Daarmee wordt aangegeven dat het nog steeds om hetzelfde data-object gaat, maar kunnen ook de verschillende statussen aangegeven worden. De specialisaties van het data-object worden dan gekoppeld aan de applicaties. De onderstaande figuur geeft dat weer.



Figuur 5-29: De status als specialisatie van een data-object

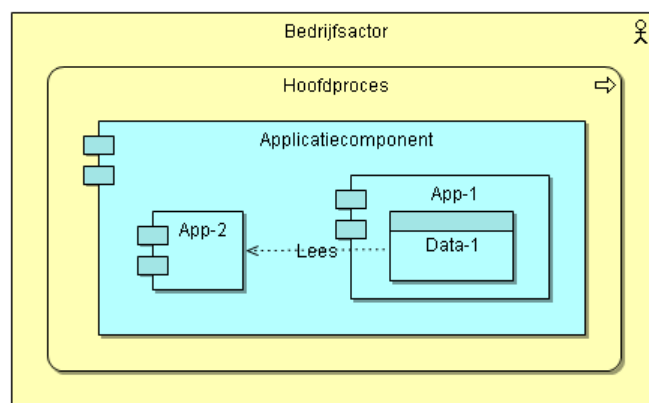
**Relaties met andere good practices:** Hier ligt een relatie met de good practice 'Onderscheid systeemsoftware en artefact'.

### 5.13 'Nesten' en impliciet afgeleide relaties

**Vraagstelling:** Wanneer ik concepten genest weergeef, gelden de relaties van het geneste object dan ook impliciet voor het 'ouder' object?

**Oplossing:** Het begrip 'nesten' is formeel niet gedefinieerd in de ArchiMatetaal. In de praktijk echter komen verschillende vormen van nesten voor:

1. Een visuele weergave van verschillende geneste concepten:



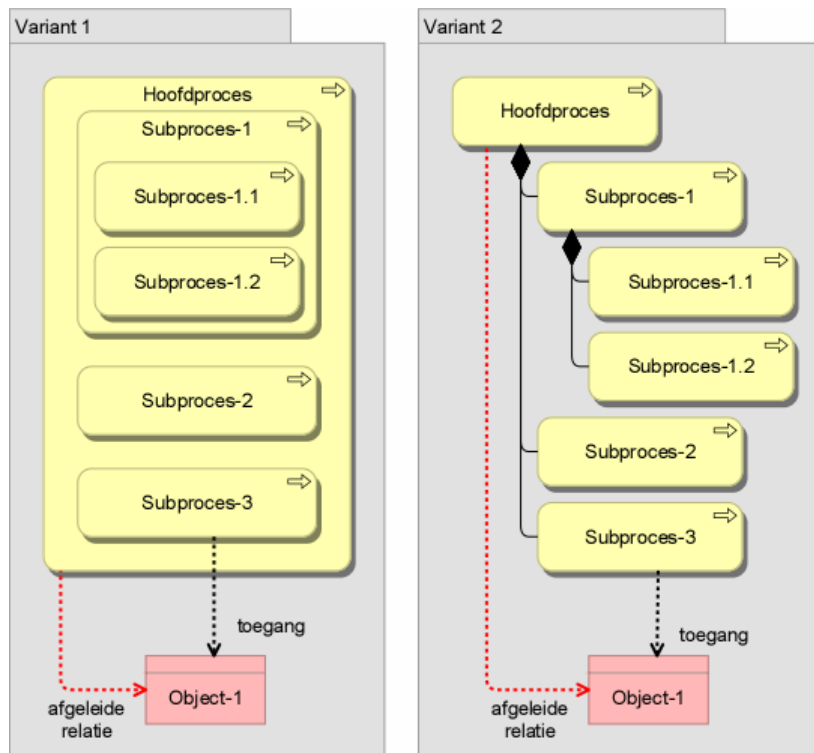
Figuur 5-30: Verschillende geneste concepten

Het voordeel van deze vorm is de relatieve eenvoudige weergave waarbij geabstraheerd is van de relaties tussen de verschillende objecten. Het nadeel is dat in deze weergave niet bekend is wat de relaties tussen de objecten betekenen. Uiteraard dienen deze relaties wel beschreven te zijn. Deze vorm van nesten wordt vaak in views gebruikt en zijn vaak een handig visueel hulpmiddel.

## 2. Hiërarchische decompositie:

Deze vorm van nesten wordt vaak gebruikt om eenzelfde concept (door middel van decompositie op te delen in deelobjecten. Voorbeelden zijn hiërarchische proces-, functie- of organisatieschema's.

Het volgende voorbeeld geeft een proceshiërarchie weer.



Figuur 5-31: Nesten via een proceshiërarchie

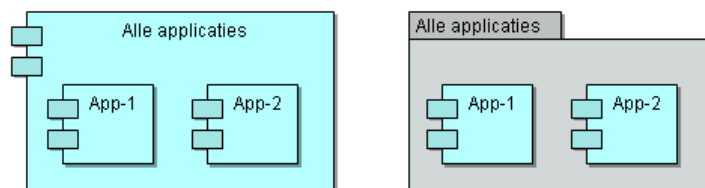
Variant 1 geeft de deelprocessen visueel genest weer. De relaties tussen de deelprocessen en het hoofdproces zijn niet zichtbaar (maar dienen formeel wel gemodelleerd te zijn).

Variant 2 geeft, in dit geval, de compositierelaties tussen de objecten weer.

Bij deze vorm van nesten is meestal sprake van compositie- of aggregatierelaties van hetzelfde concept.

In de beide varianten is tevens gemodelleerd dat Subproces-3 een toegangsrelatie heeft met Object-1. Omdat subproces-3 een decompositie is van het Hoofdproces geldt volgens de regels van afgeleide relaties (zie good practice 'Vereenvoudigingen met behoud van consistentie') tevens dat het Hoofdproces een toegangsrelatie heeft met Object-1. En dus geldt dat bij deze vorm van nesten het 'ouderobject' dezelfde relaties heeft als het 'kindobject'.

Noot: Compositie of decompositie van een concept is iets anders dan een clustering van gelijksoortige concepten. Voor dit laatste kan het concept 'Groep' gebruikt worden.



Figuur 5-32: Clustering via 'Groep'

**Gevolgen:** Bij het 'visueel' nesten van objecten dient expliciet de relaties tussen de objecten beschreven te worden.

**Alternatieven:** Niet van toepassing.

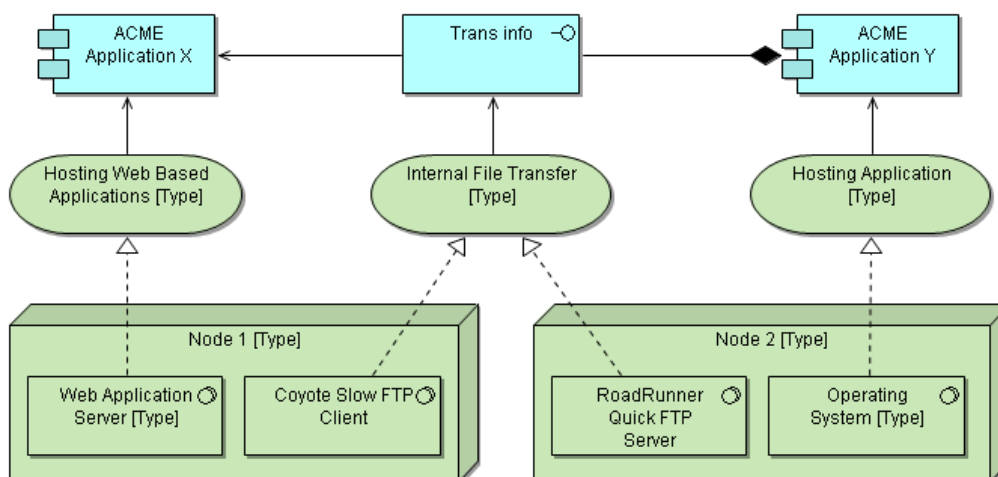
**Relaties met andere good practices:** Zie ook de good practice 'Vereenvoudigingen met behoud van consistentie'.

#### 5.14 File transfer

**Vraagstelling:** Hoe kan ik file transfer (FTP e.d.) modelleren?

**Oplossing:** Er zijn diverse manieren om File Transfer af te beelden met behulp van ArchiMate. Uitgangspunt voor deze good practice is het technology usage viewpoint. Dit is een viewpoint waarin wordt aangegeven welke infrastructuurservices worden gebruikt door bepaalde applicaties.

File Transfer is een infrastructuurservice die wordt gebruikt door twee applicaties: een zender en een ontvanger. Om deze service te kunnen realiseren moet de hosting omgeving van zowel de zender als de ontvanger zijn voorzien van systeemsoftware die deze infrastructuurfunctionaliteit ondersteunt.



Figuur 5-33: File transfer

In bovenstaande view zien we dat er tussen applicatie X en Y een applicatie-interface is, genaamd 'Trans info'. Applicatie Y heeft deze interface en Applicatie X gebruikt deze interface. Dit wordt weergegeven middels een compositie en gebruikt door relatie. De applicatie-interface maakt gebruik van de infrastructuurservice 'Internal File Transfer'. De service wordt gerealiseerd door enerzijds 'Coyote Slow FTP client' systeemsoftware op de node waar ook de hosting van applicatie X plaatsvindt en anderzijds door 'RoadRunner Quick FTP server' systeemsoftware op de hosting omgeving van applicatie Y.

In een technology usage view is de richting van de flow geen doel om uit te beelden. Impliciet kan uit bovenstaande view worden afgeleid dat de flow plaatsvindt van X naar Y. Zou er ook een flow van Y naar X bestaan, dan moet hiervoor een andere applicatie interface gedefinieerd worden.

**Gevolgen:** Om deze good practice te kunnen gebruiken moeten flowrelaties tussen applicaties vervangen worden door applicatie-interfaces. Aan een flowrelatie kan namelijk geen gebruikt door relatie gekoppeld worden.

**Alternatieven:** Niet van toepassing.

**Relaties met andere good practices:** Hier ligt een relatie met de good practice 'Weergeven van berichten in wachtrijen'. Een soortgelijke good practice zou ook gebruikt kunnen worden voor infrastructuurservices zoals Messaging. In dat geval zal de 'Systeemsoftware' niet op FTP gebaseerd zijn maar bijv. op MQ of RV.

## Referenties

- M. Lankhorst, *Enterprise Architecture At Work – Modelling, Communication, and Analysis*, Telematica Instituut/Springer, 2005, ISBN 3-540-24371-2.
- H. Bosma, H. Jonkers en M. Lankhorst, *Inleiding in de ArchiMate-taal*, ArchiMate/D.1.1.6a, v1.1, Telematica Instituut/Ordina, 24 oktober 2005.  
<https://doc.telin.nl/dscgi/ds.py/Get/File-49772>.
- R. van Buuren, S. Hoppenbrouwers, H. Jonkers, M. Lankhorst en G. Veldhuijzen van Zanten, *Architecture Language Reference Manual*, ArchiMate/D2.2.2b, v4.1, Telematica Instituut, 4 april 2006. <https://doc.telin.nl/dscgi/ds.py/Get/File-31626>.
- H. ter Doest, M.-E. Iacob, M. Lankhorst, D. van Leeuwen en R. Slagter, *Viewpoints Functionality and Examples*, ArchiMate/D3.4.1a, v2.6, Telematica Instituut, 18 november 2004. <https://doc.telin.nl/dscgi/ds.py/Get/File-35434>.
- ArchiMate Team, *ArchiMate Quick Reference*, Telematica Instituut, 2005.  
<https://doc.telin.nl/dscgi/ds.py/Get/File-52048>
- The Open Group, *ArchiMate 1.0 Specification – Technical Standard*, The Open Group, 2009. <http://www.opengroup.org/bookstore/catalog/c091.htm>.
- The Open Group, *ArchiMate 2.0 Specification – Technical Standard*, The Open Group, 2012. <http://www.opengroup.org/bookstore/catalog/c118.htm>.
- M.M. Lankhorst, A.J. Klievink, P.H.W.M. Oude Luttighuis, E. Felt, L. Heerink, D. van Leeuwen, *Kanaalpatronen – Kanalen in Balans*, Novay / TU Delft, 2009.  
<https://doc.novay.nl/dsweb/Get/Document-88739/D3.2%20Kanaalpatronen.pdf>.